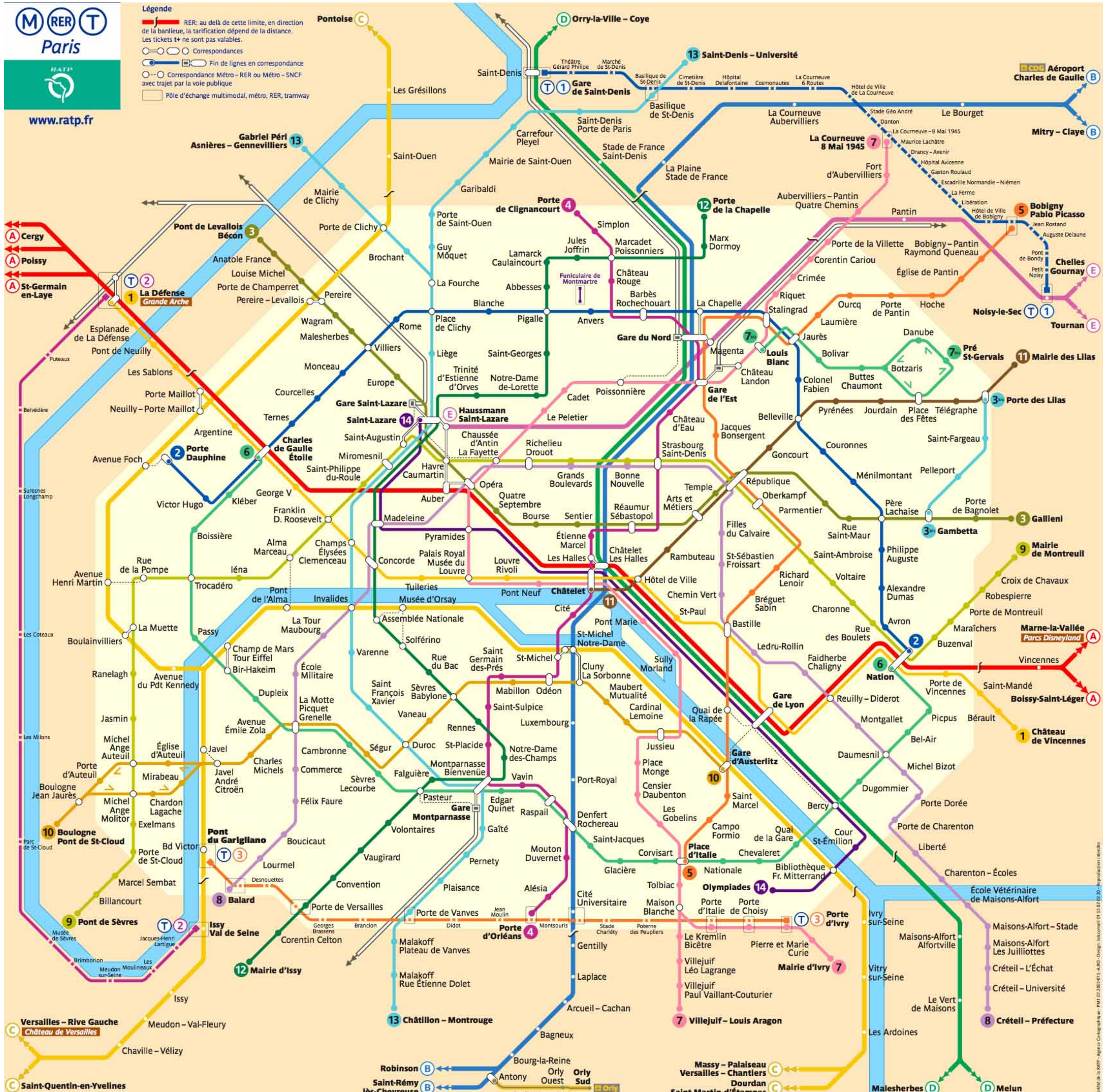
# Metro-Line Crossing Minimization: Hardness, Approximations, and Tractable Cases

Martin Fink
Lehrstuhl für Informatik I
Universität Würzburg

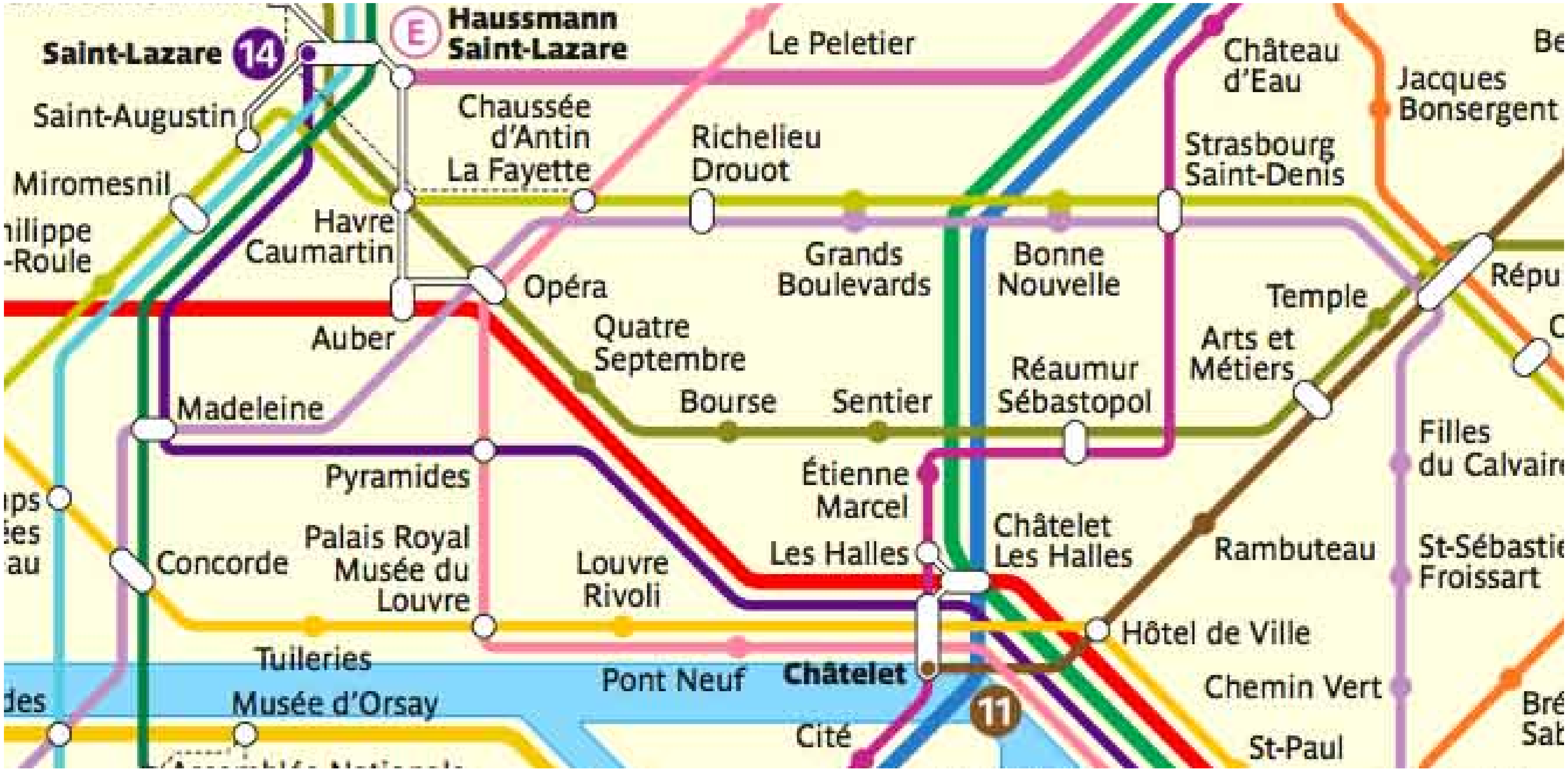
Joint work with Sergey Pupyrev

# Metro Maps – Bordeaux

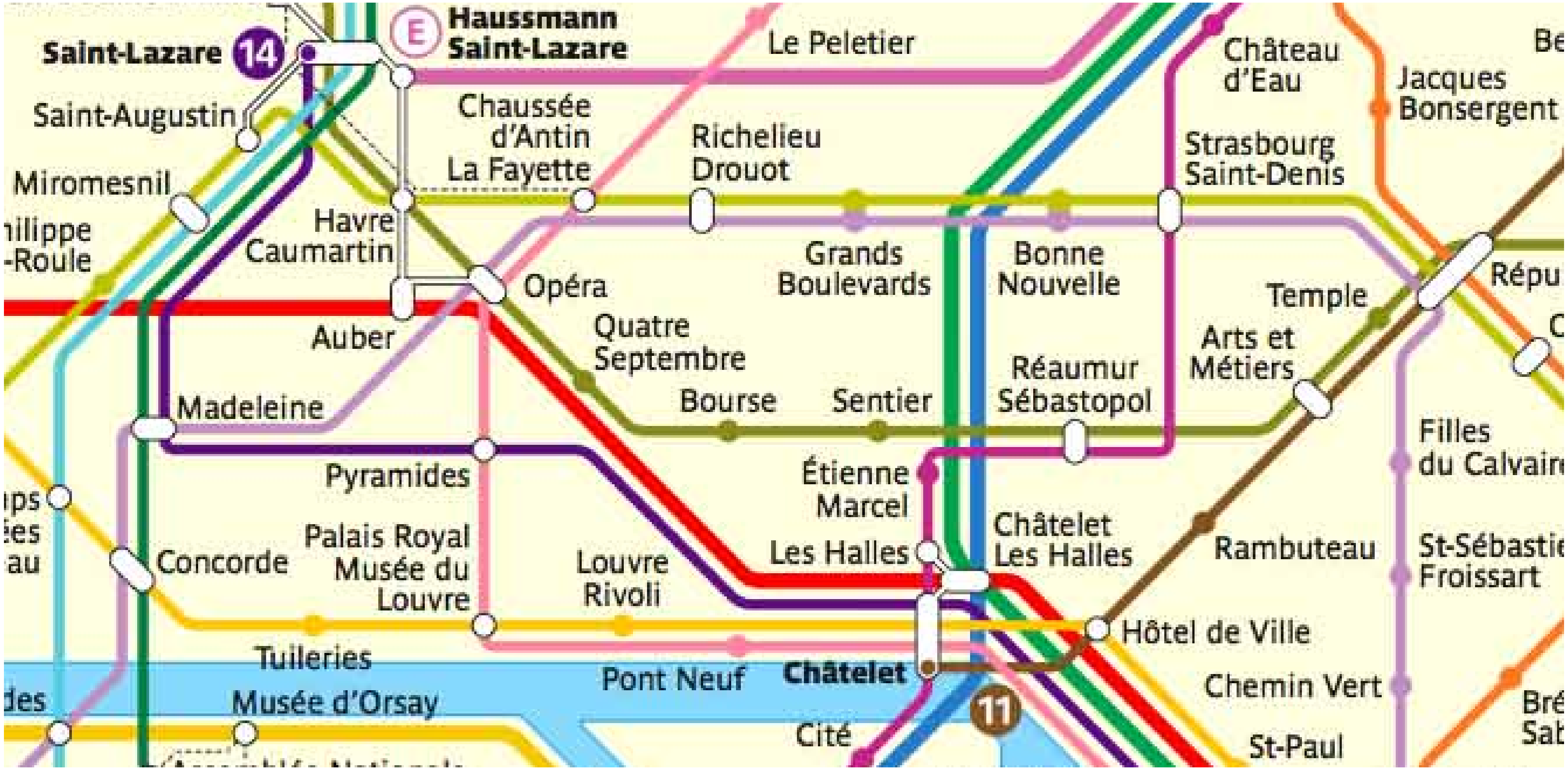
# Metro Maps – Paris



# Metro Maps – Metro Lines



# Metro Maps – Metro Lines



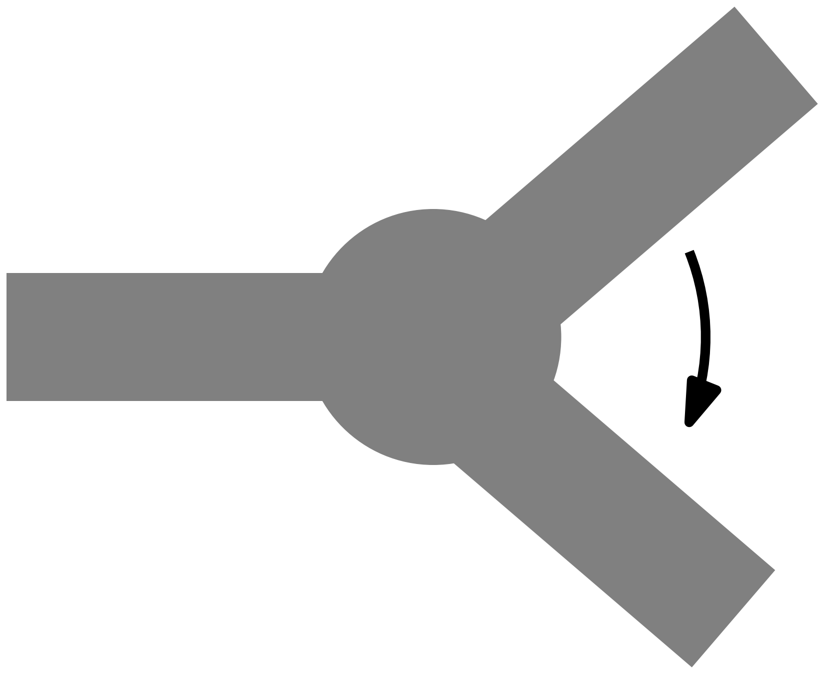

Previous work, e.g. [Nöllenburg and Wolff, 2011]
with focus on drawing underlying graph

# Metro Line Crossing Minimization

Insert all lines $L$ into *embedded graph* $G = (V, E)$ such that ...

# Metro Line Crossing Minimization

Insert all lines $L$ into *embedded graph* $G = (V, E)$ such that ...



# Metro Line Crossing Minimization

Insert all lines $L$ into *embedded graph* $G = (V, E)$ such that ...

- the number of crossings is minimized.

# Metro Line Crossing Minimization

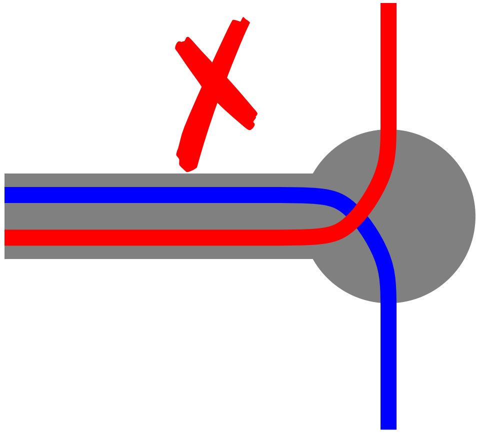
Insert all lines $L$ into *embedded graph* $G = (V, E)$ such that ...

- the number of crossings is minimized.
- crossings are placed on edges if possible.

# Metro Line Crossing Minimization

Insert all lines $L$ into *embedded graph* $G = (V, E)$ such that ...

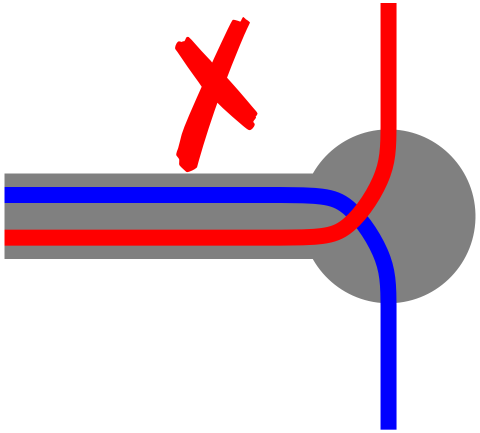
- the number of crossings is minimized.
- crossings are placed on edges if possible.

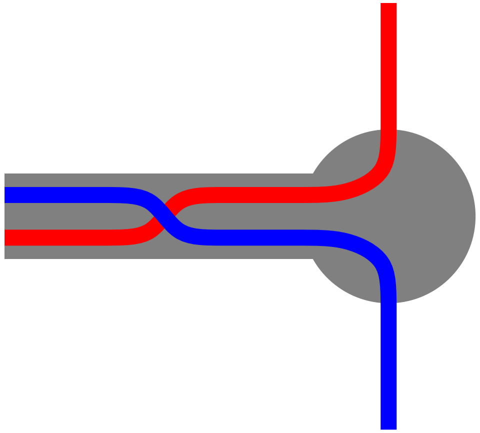


# Metro Line Crossing Minimization

Insert all lines $L$ into *embedded graph* $G = (V, E)$ such that ...

- the number of crossings is minimized.
- crossings are placed on edges if possible.

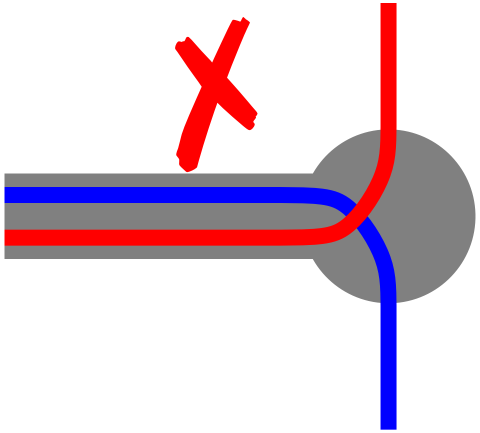


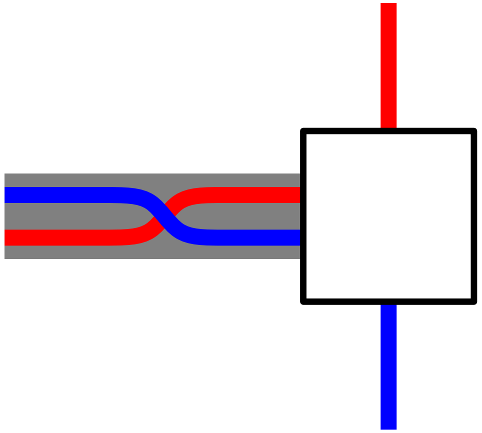


# Metro Line Crossing Minimization

Insert all lines $L$ into *embedded graph* $G = (V, E)$ such that ...

- the number of crossings is minimized.
- crossings are placed on edges if possible.





# Metro Line Crossing Minimization

Insert all lines $L$ into *embedded graph* $G = (V, E)$ such that ...

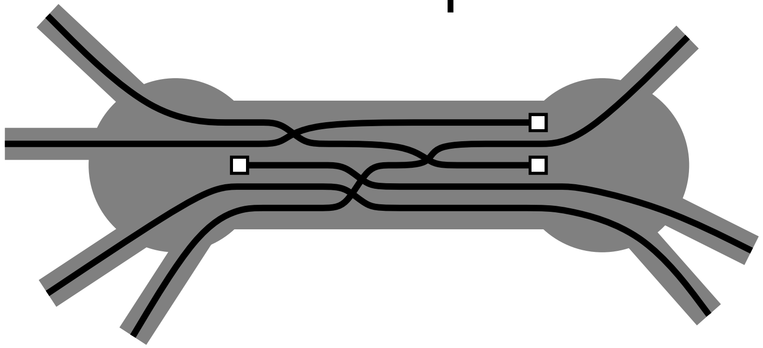
- the number of crossings is minimized.
- crossings are placed on edges if possible.

We assume that . . .

- lines are simple paths.
- lines intersect in paths.

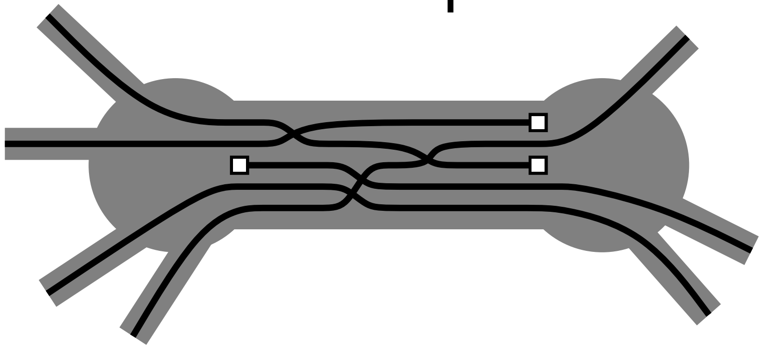
# Problem Variants

- **MLCM**: line ends placed freely

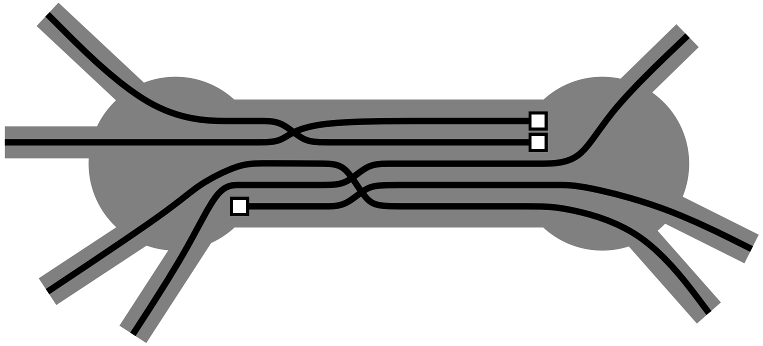


# Problem Variants

- **MLCM**: line ends placed freely

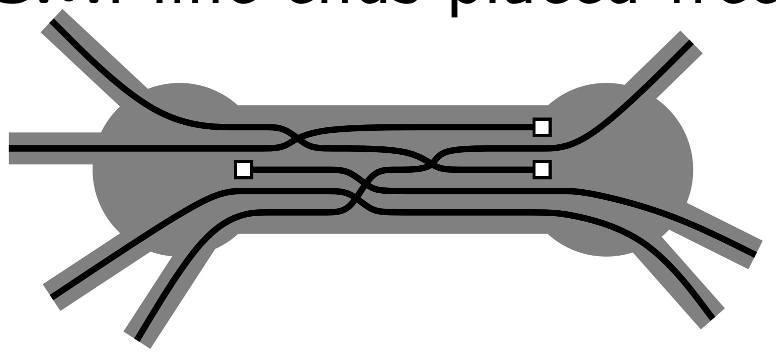


- **MLCM-P**: line ends placed outermost

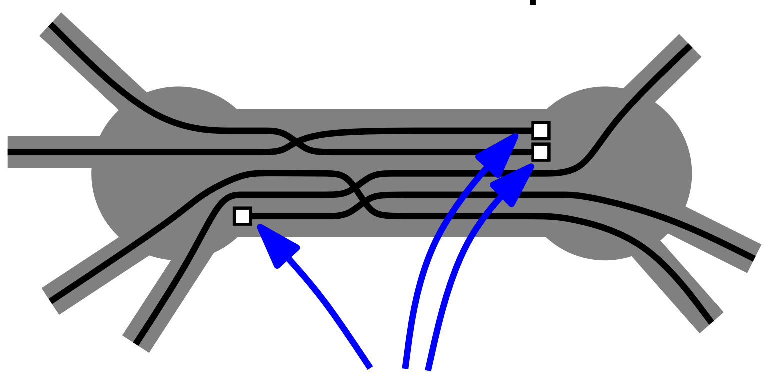


# Problem Variants

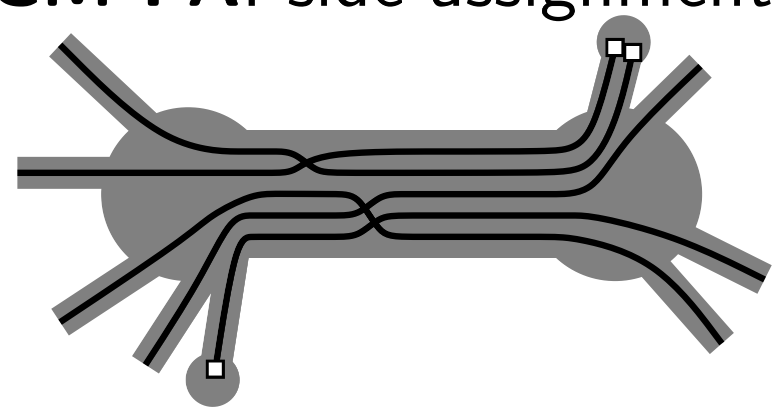
- **MLCM**: line ends placed freely



- **MLCM-P**: line ends placed outermost



- **MLCM-PA**: side assignment for line ends



equivalent: line ends in degree-1 vertices

# Previous Work

- **MLCM**: single edge: $O(|L|^2)$ time [Benkert et al., GD'06]
  more complex graphs: complexity open

# Previous Work

- **MLCM**: single edge: $O(|L|^2)$ time [Benkert et al., GD'06]
  more complex graphs: complexity open
- **MLCM-P**:
  – NP-hard on paths [Bekos et al., GD'07]

# Previous Work

- **MLCM**: single edge: $O(|L|^2)$ time [Benkert et al., GD'06]
  more complex graphs: complexity open
- **MLCM-P**:
  – NP-hard on paths [Bekos et al., GD'07]
  – solution by ILP [Asquith et al., 2008]

# Previous Work

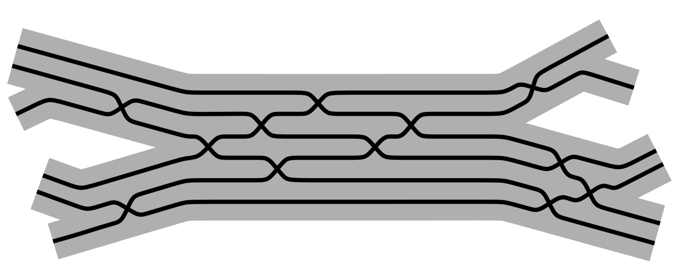
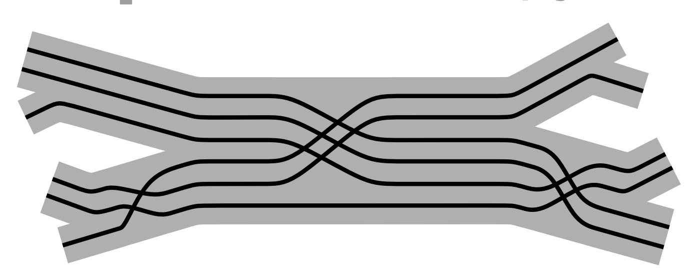
- **MLCM**: single edge: $O(|L|^2)$ time [Benkert et al., GD'06]
  more complex graphs: complexity open
- **MLCM-P**:
  - NP-hard on paths [Bekos et al., GD'07]
  - solution by ILP [Asquith et al., 2008]
- **MLCM-PA**:
  - polynomial-time algorithms by [Asquith et al., 2008], [Argyriou et al., GD'08], and [Nöllenburg, GD'09]

# Previous Work

- **MLCM**: single edge: $O(|L|^2)$ time [Benkert et al., GD'06]
  more complex graphs: complexity open
- **MLCM-P**:
  – NP-hard on paths [Bekos et al., GD'07]
  – solution by ILP [Asquith et al., 2008]
- **MLCM-PA**:
  – polynomial-time algorithms by [Asquith et al., 2008], [Argyriou et al., GD'08], and [Nöllenburg, GD'09]
  – linear-time algorithm [Pupyrev et al., GD'11]

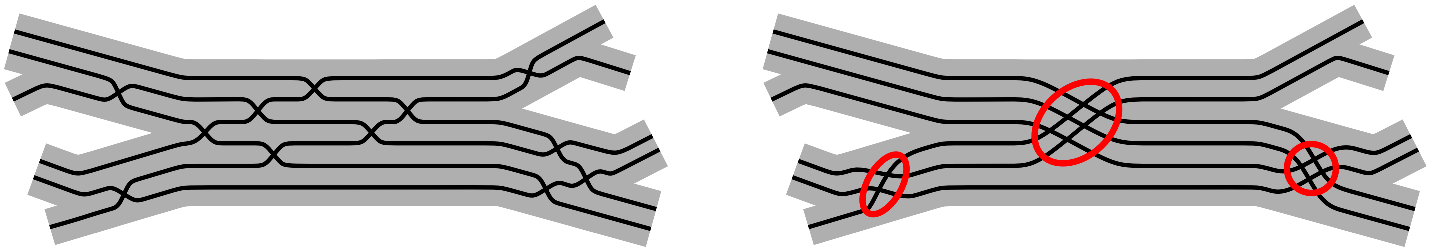
# Previous Work

- **MLCM**: single edge: $O(|L|^2)$ time [Benkert et al., GD'06]
  more complex graphs: complexity open
- **MLCM-P**:
  – NP-hard on paths [Bekos et al., GD'07]
  – solution by ILP [Asquith et al., 2008]
- **MLCM-PA**:
  – polynomial-time algorithms by [Asquith et al., 2008], [Argyriou et al., GD'08], and [Nöllenburg, GD'09]
  – linear-time algorithm [Pupyrev et al., GD'11]
  – "block crossings" [Fink and Pupyrev, 2013]

# Previous Work

- **MLCM**: single edge: $O(|L|^2)$ time [Benkert et al., GD'06]
  more complex graphs: complexity open
- **MLCM-P**:
  - NP-hard on paths [Bekos et al., GD'07]
  - solution by ILP [Asquith et al., 2008]
- **MLCM-PA**:
  - polynomial-time algorithms by [Asquith et al., 2008], [Argyriou et al., GD'08], and [Nöllenburg, GD'09]
  - linear-time algorithm [Pupyrev et al., GD'11]
  - "block crossings" [Fink and Pupyrev, 2013]



# Our results

- Main results:
  - **MLCM** is NP-hard.
  - $O(\sqrt{\log |L|})$-approximation for **MLCM-P**

# Our results

- Main results:
  - **MLCM** is NP-hard.
  - $O(\sqrt{\log |L|})$-approximation for **MLCM-P**
- Additional results:
  - **MLCM** $+$ **MLCM-P**: check for crossing-free solution

# Our results

- Main results:
  - **MLCM** is NP-hard.
  - $O(\sqrt{\log |L|})$-approximation for **MLCM-P**
- Additional results:
  - **MLCM** + **MLCM-P**: check for crossing-free solution
  - **MLCM-P**: Checking for solution with $\leq k$ crossings is fixed-parameter tractable.

# Our results

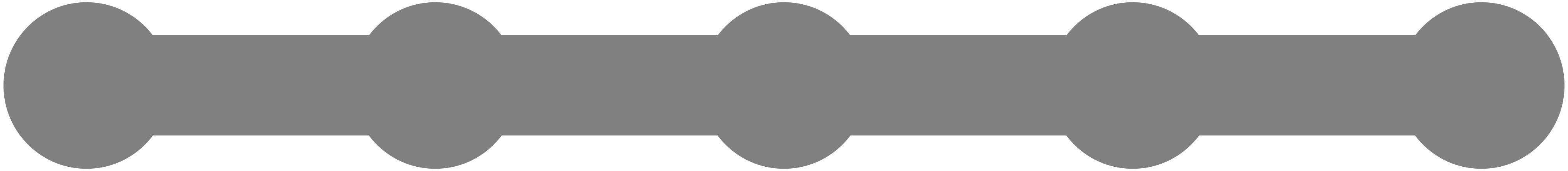
- Main results:
  - **MLCM** is NP-hard.
  - $O(\sqrt{\log|L|})$-approximation for **MLCM-P**
- Additional results:
  - **MLCM** + **MLCM-P**: check for crossing-free solution
  - **MLCM-P**: Checking for solution with $\leq k$ crossings is fixed-parameter tractable.
  - a restricted version of MLCM-P can be solved efficiently

# Our results

- Main results:
  - **MLCM** is NP-hard.
  - $O(\sqrt{\log |L|})$-approximation for **MLCM-P**
- Additional results:
  - **MLCM** + **MLCM-P**: check for crossing-free solution
  - **MLCM-P**: Checking for solution with $\leq k$ crossings is fixed-parameter tractable.
  - a restricted version of MLCM-P can be solved efficiently

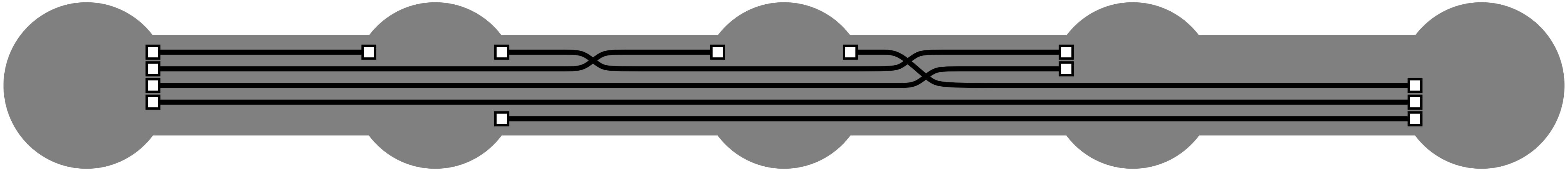
# MLCM – NP-hardness

- MLCM-P is hard on paths.



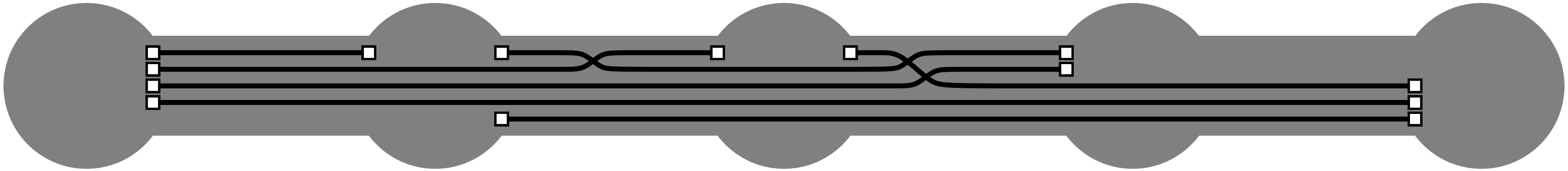
# MLCM – NP-hardness

- MLCM-P is hard on paths.

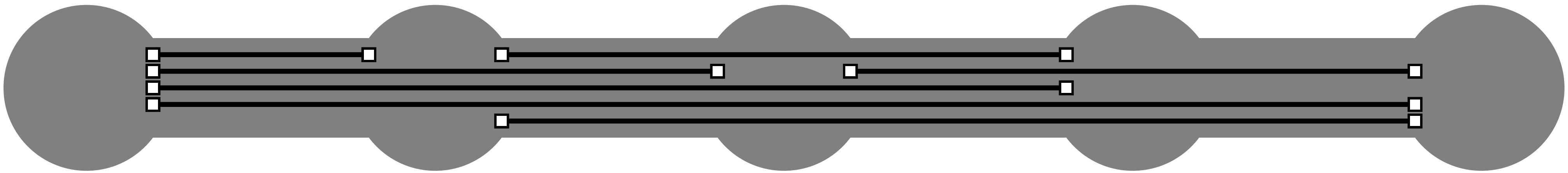


# MLCM – NP-hardness

- MLCM-P is hard on paths.

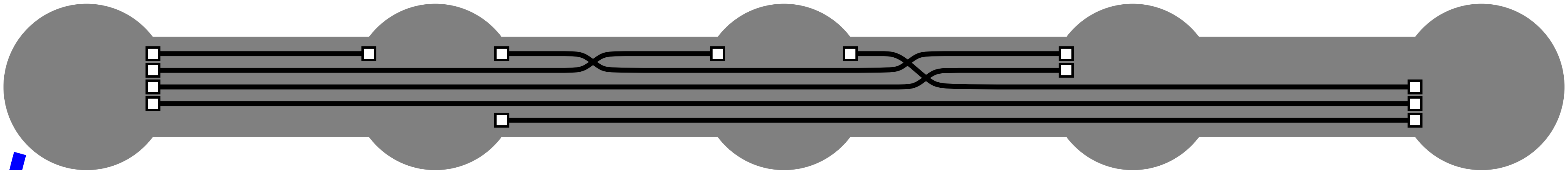


- MLCM is trivial on paths.

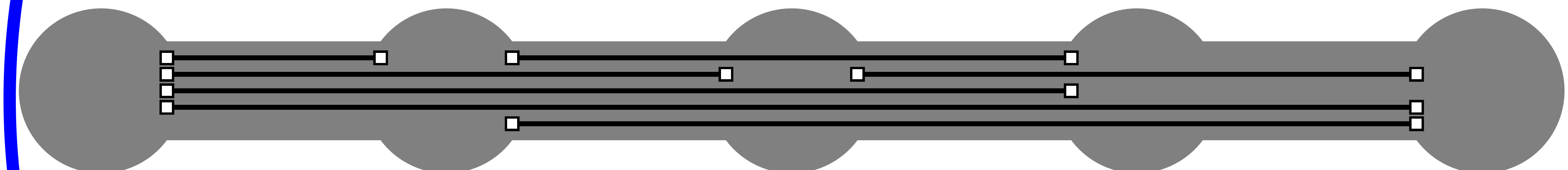


# MLCM – NP-hardness

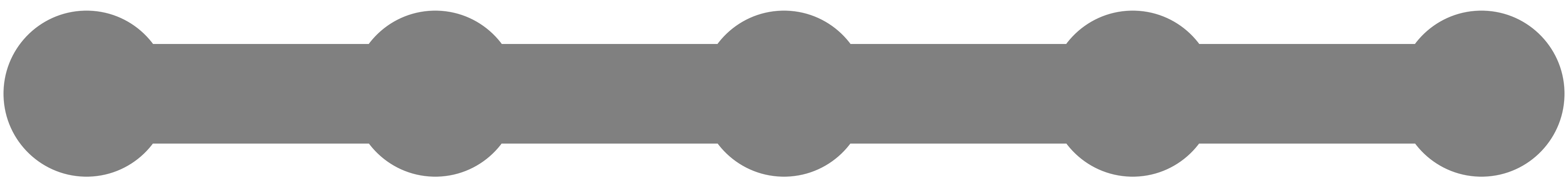
- MLCM-P is hard on paths.



- MLCM is trivial on paths.

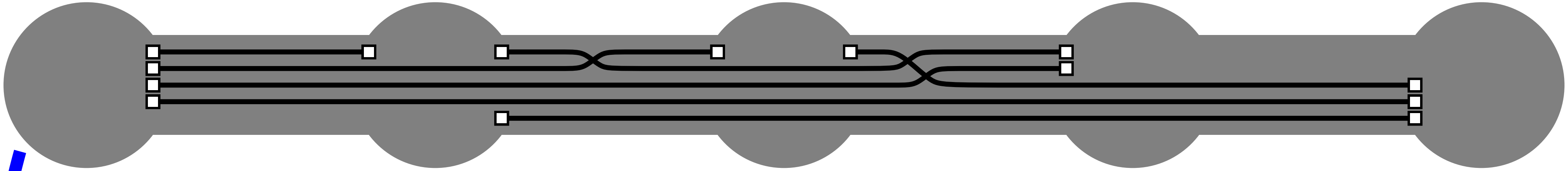


- Show hardness on caterpillars by reduction from MLCM-P.

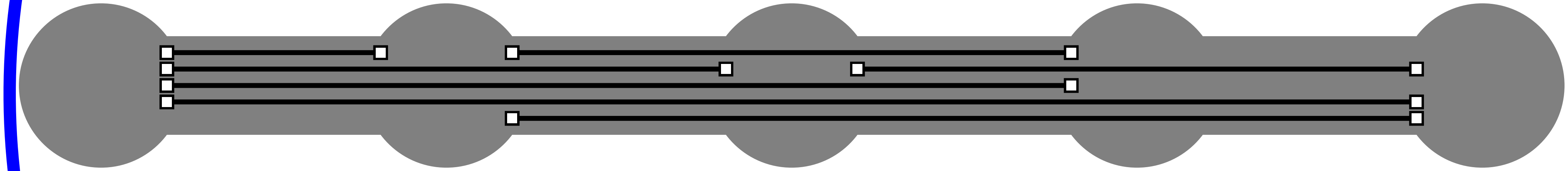


# MLCM – NP-hardness

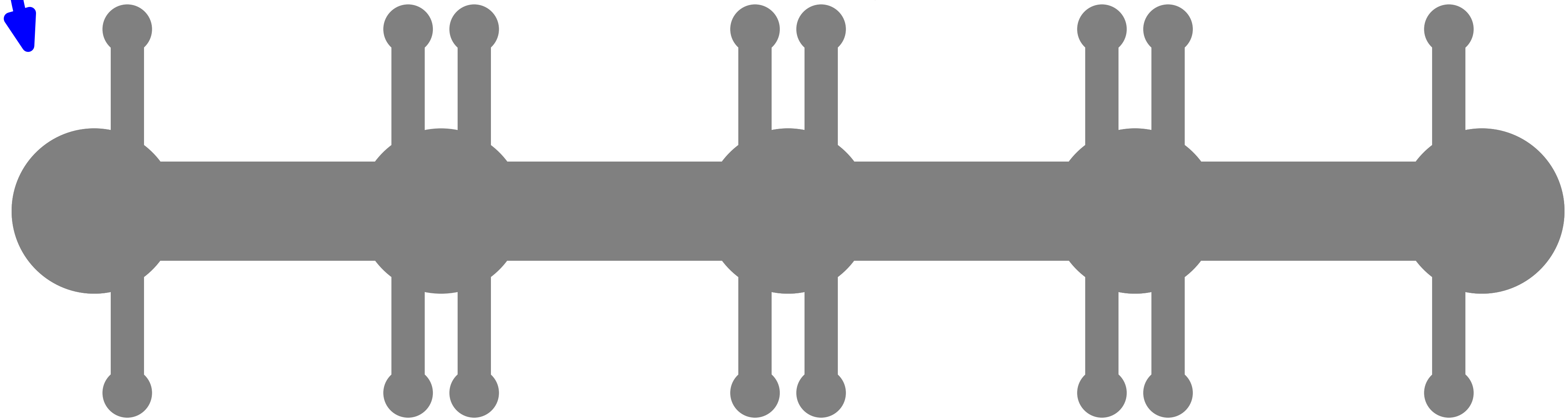
- MLCM-P is hard on paths.



- MLCM is trivial on paths.

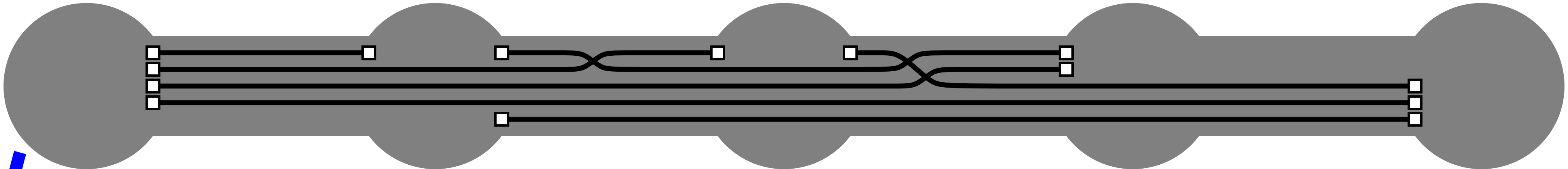


- Show hardness on caterpillars by reduction from MLCM-P.

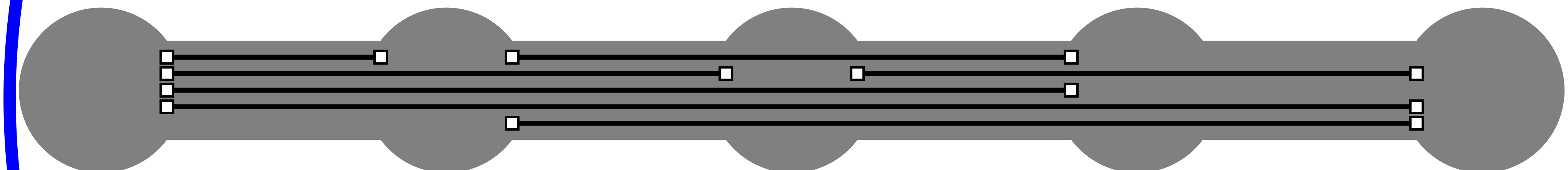


# MLCM – NP-hardness

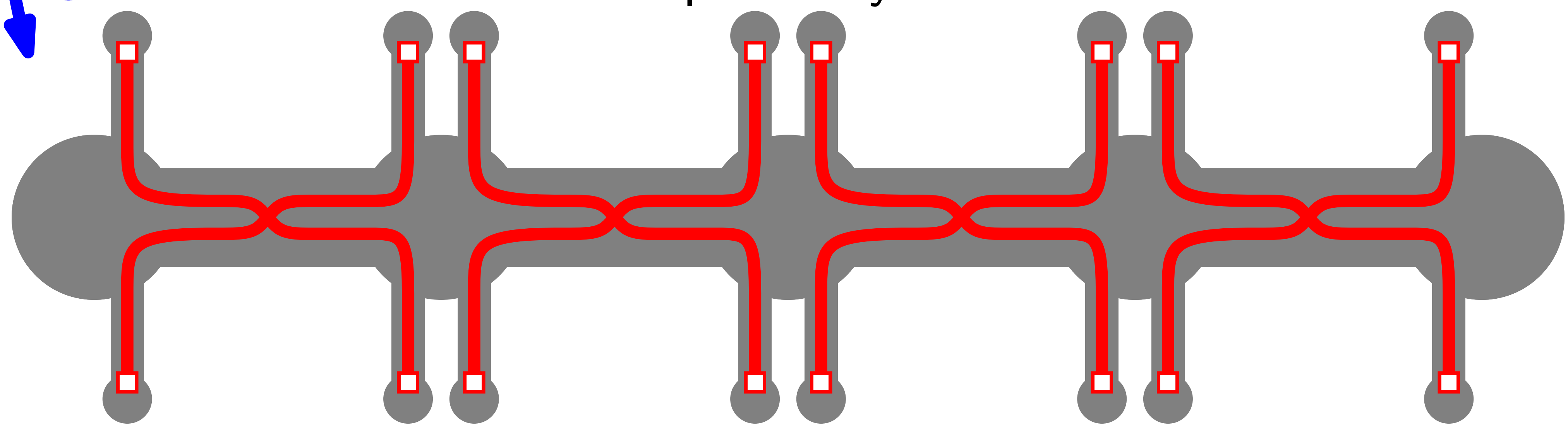
- MLCM-P is hard on paths.



- MLCM is trivial on paths.

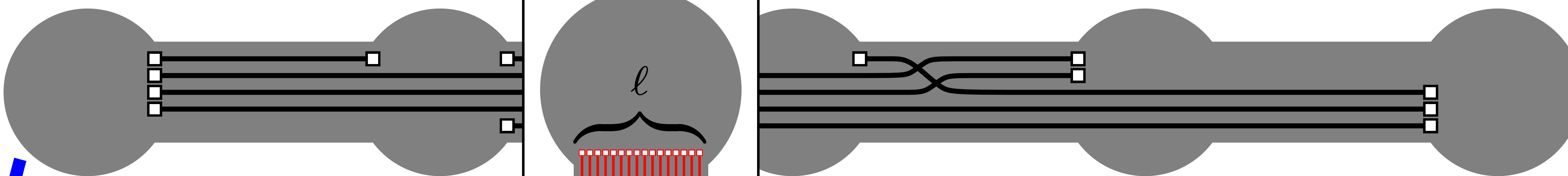


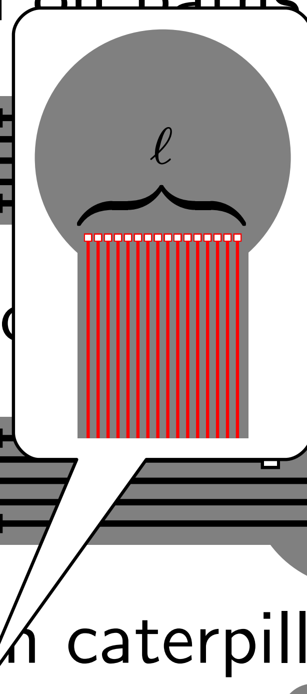
- Show hardness on caterpillars by reduction from MLCM-P.



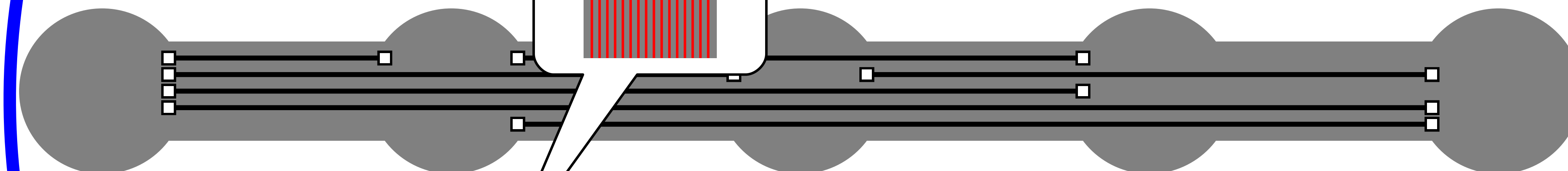
# MLCM – NP-hardness

- MLCM-P is hard on paths.

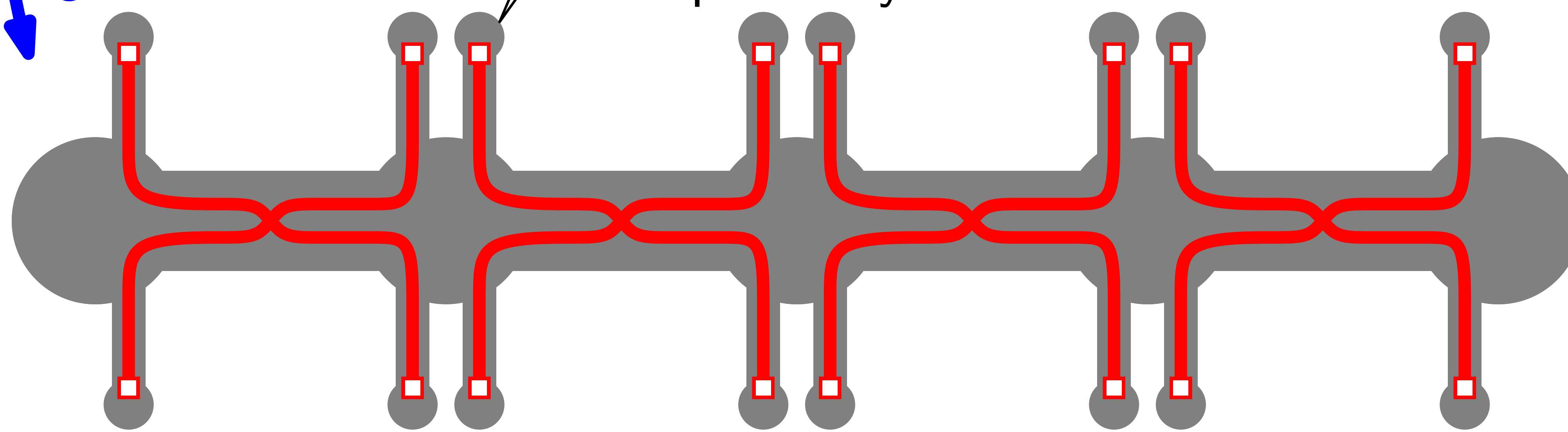




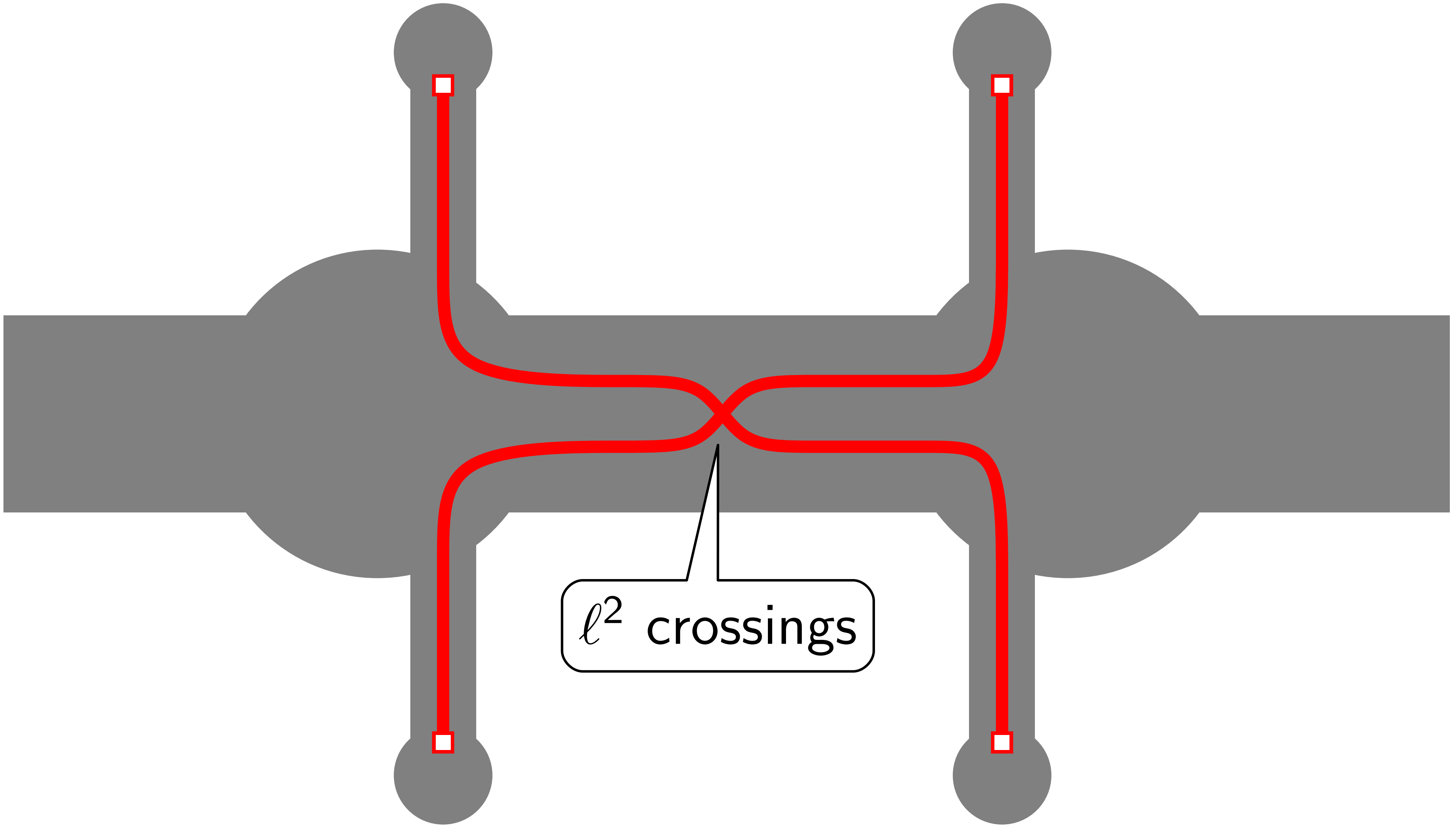

- MLCM is trivial



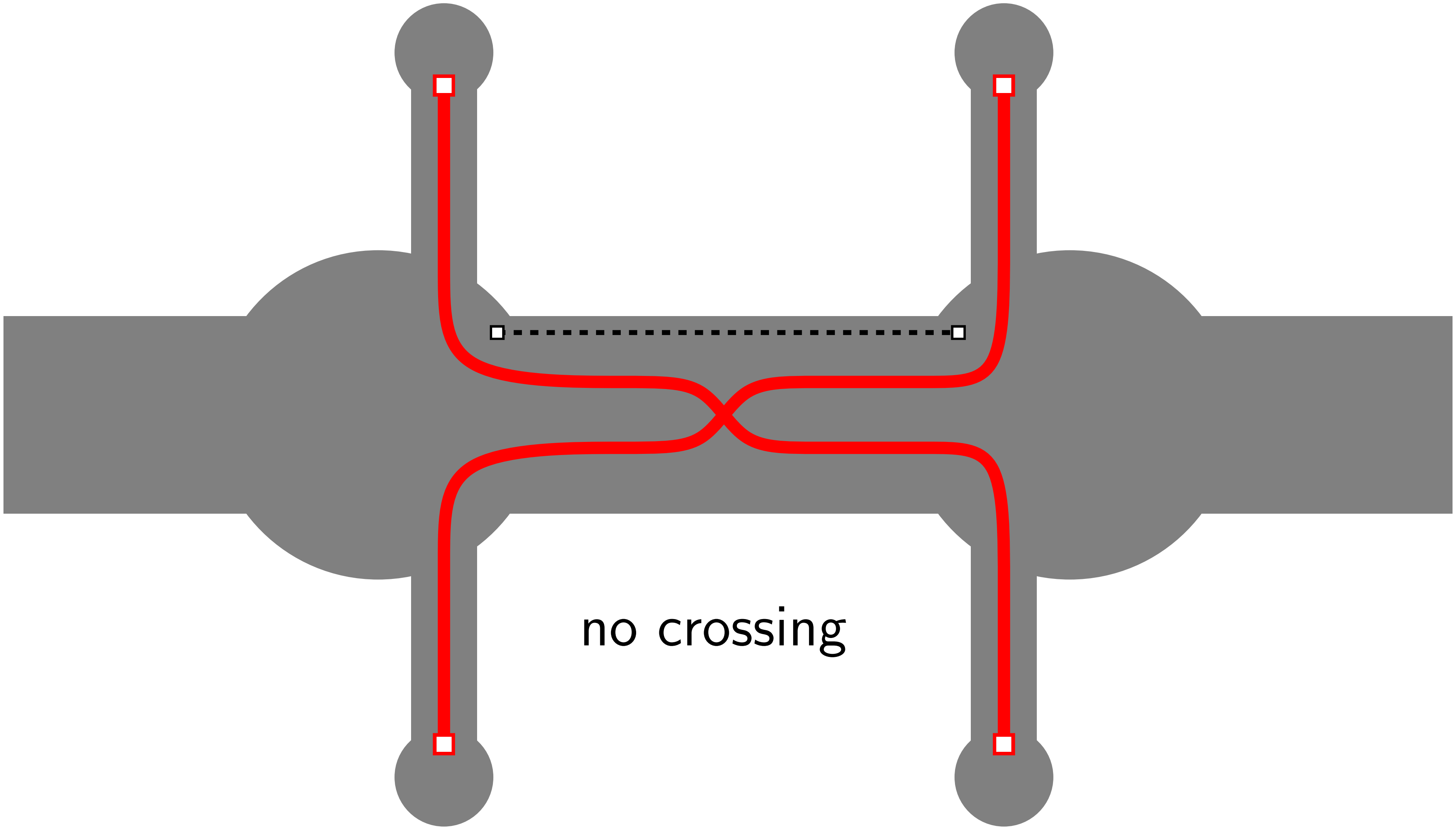
- Show hardness on caterpillars by reduction from MLCM-P.



# MLCM – NP-hardness cont'd

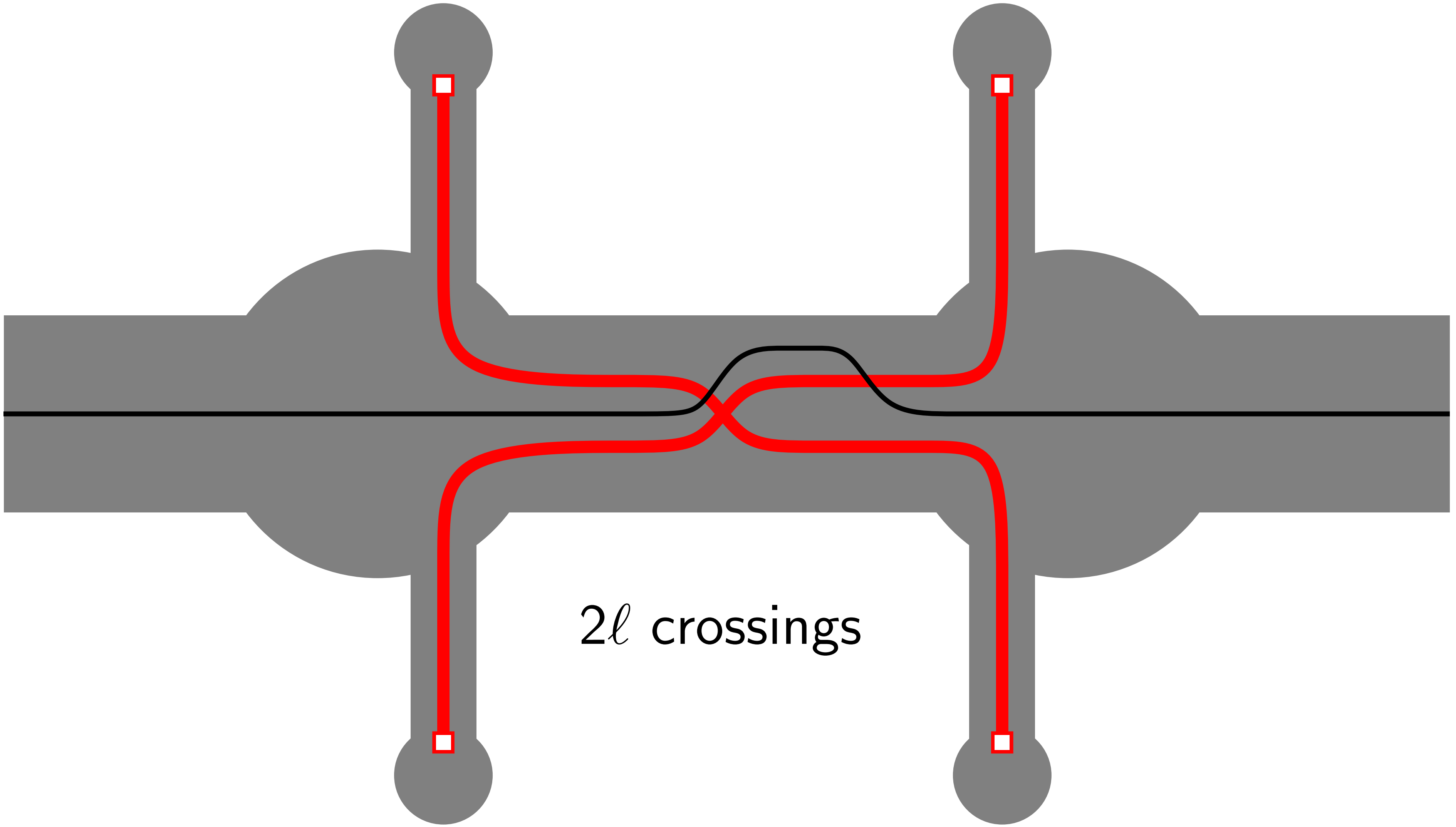


# MLCM – NP-hardness cont'd



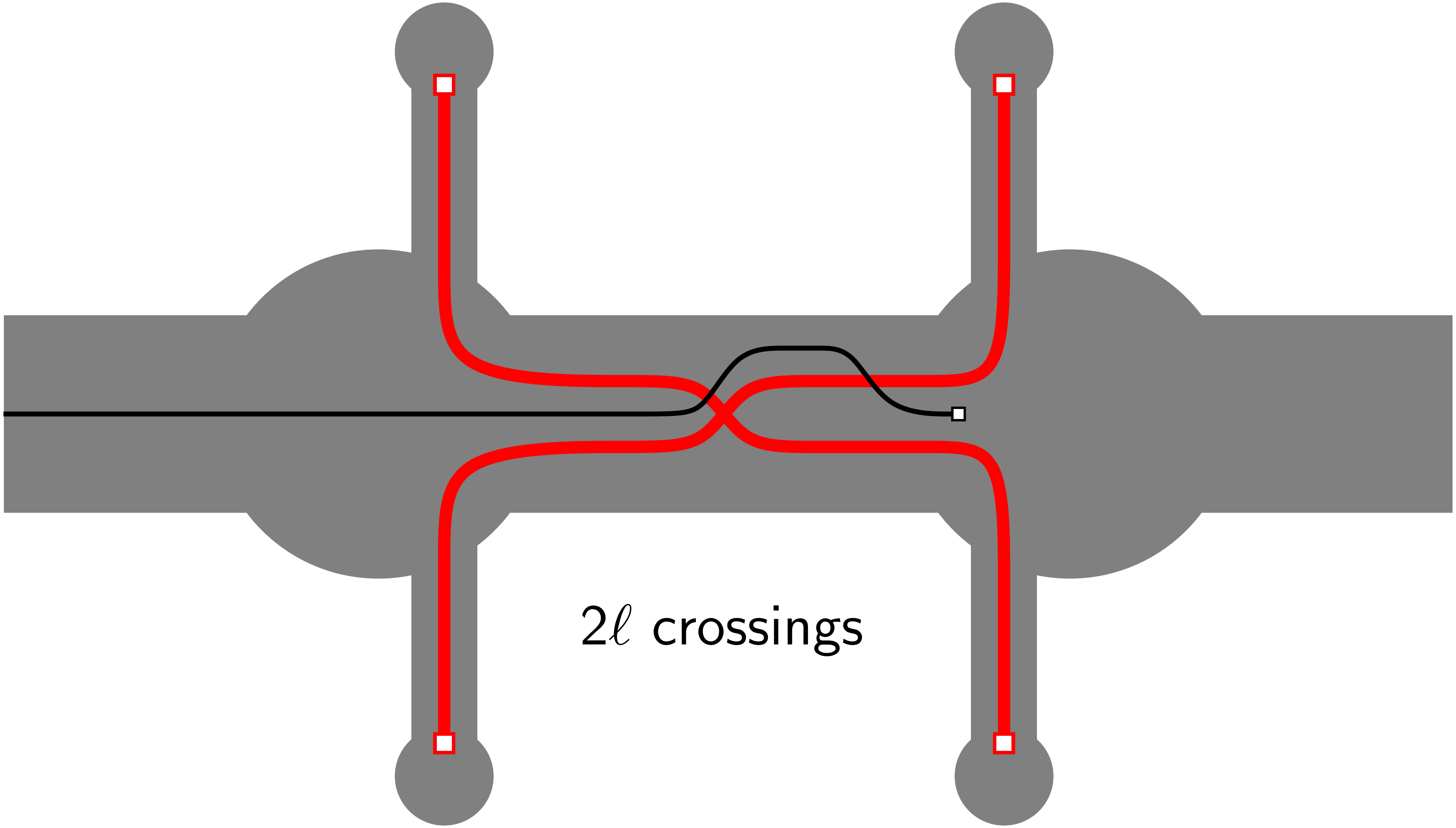
no crossing

# MLCM – NP-hardness cont'd



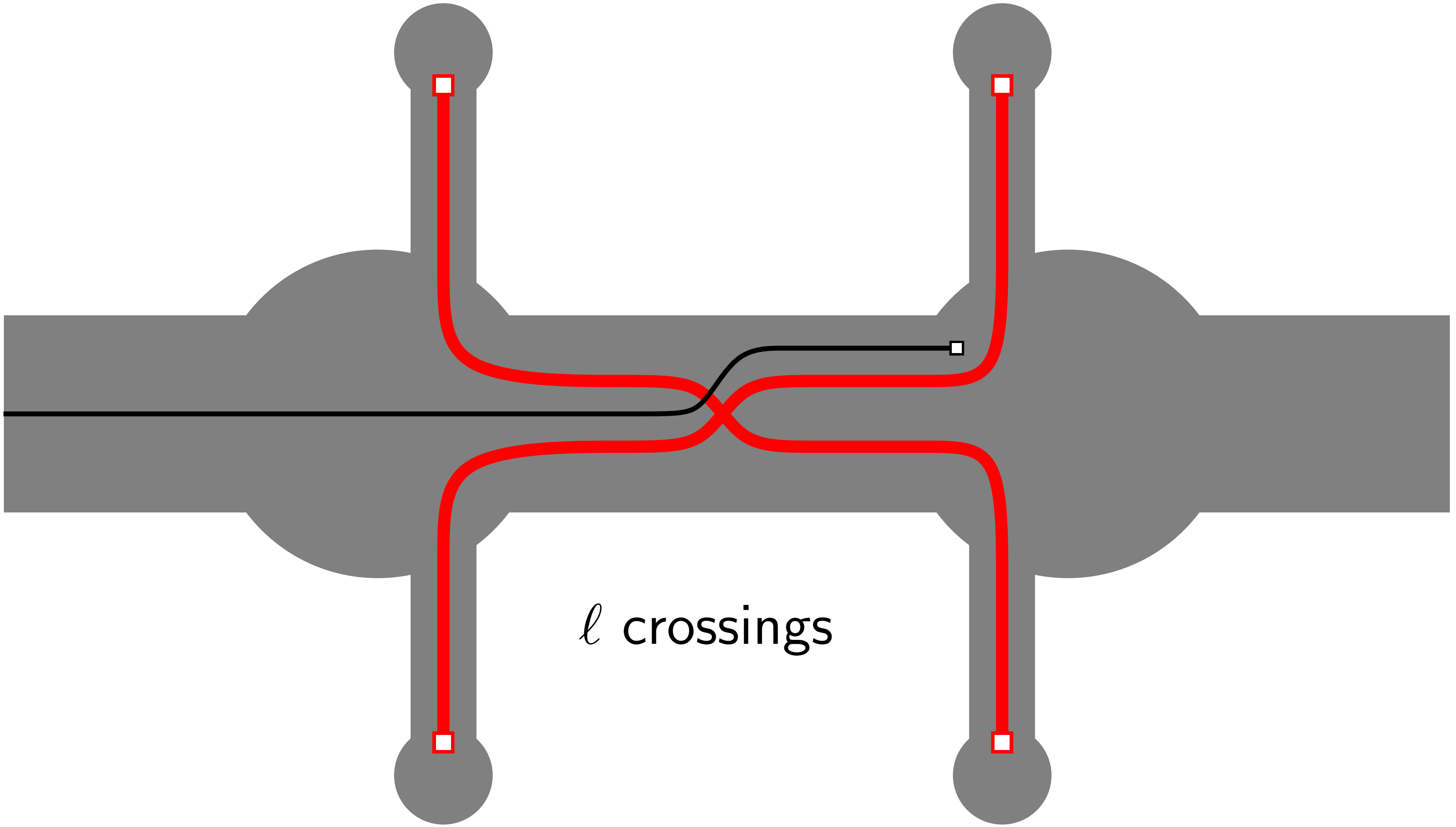
2$\ell$ crossings

# MLCM – NP-hardness cont'd

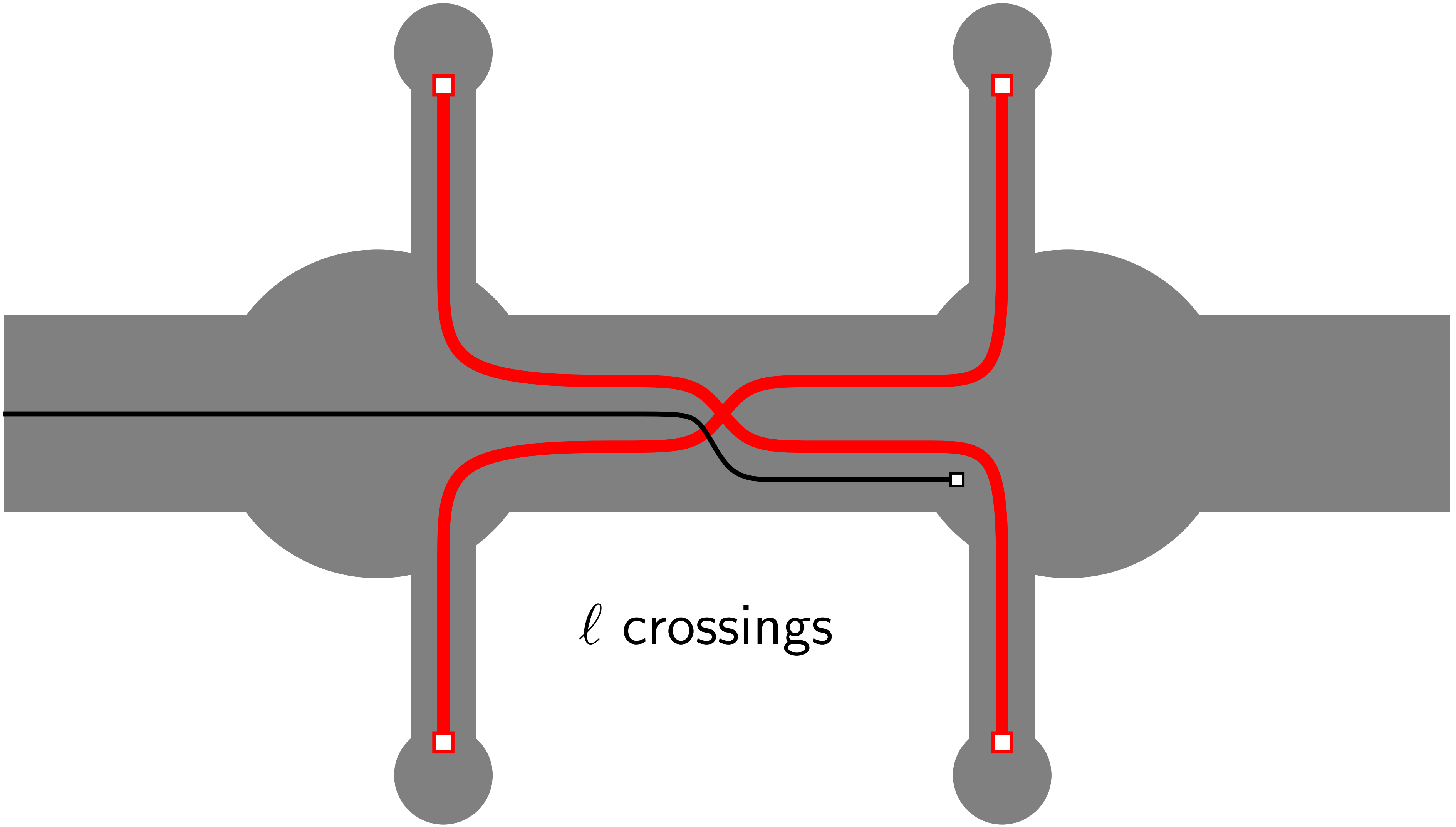


$2\ell$ crossings

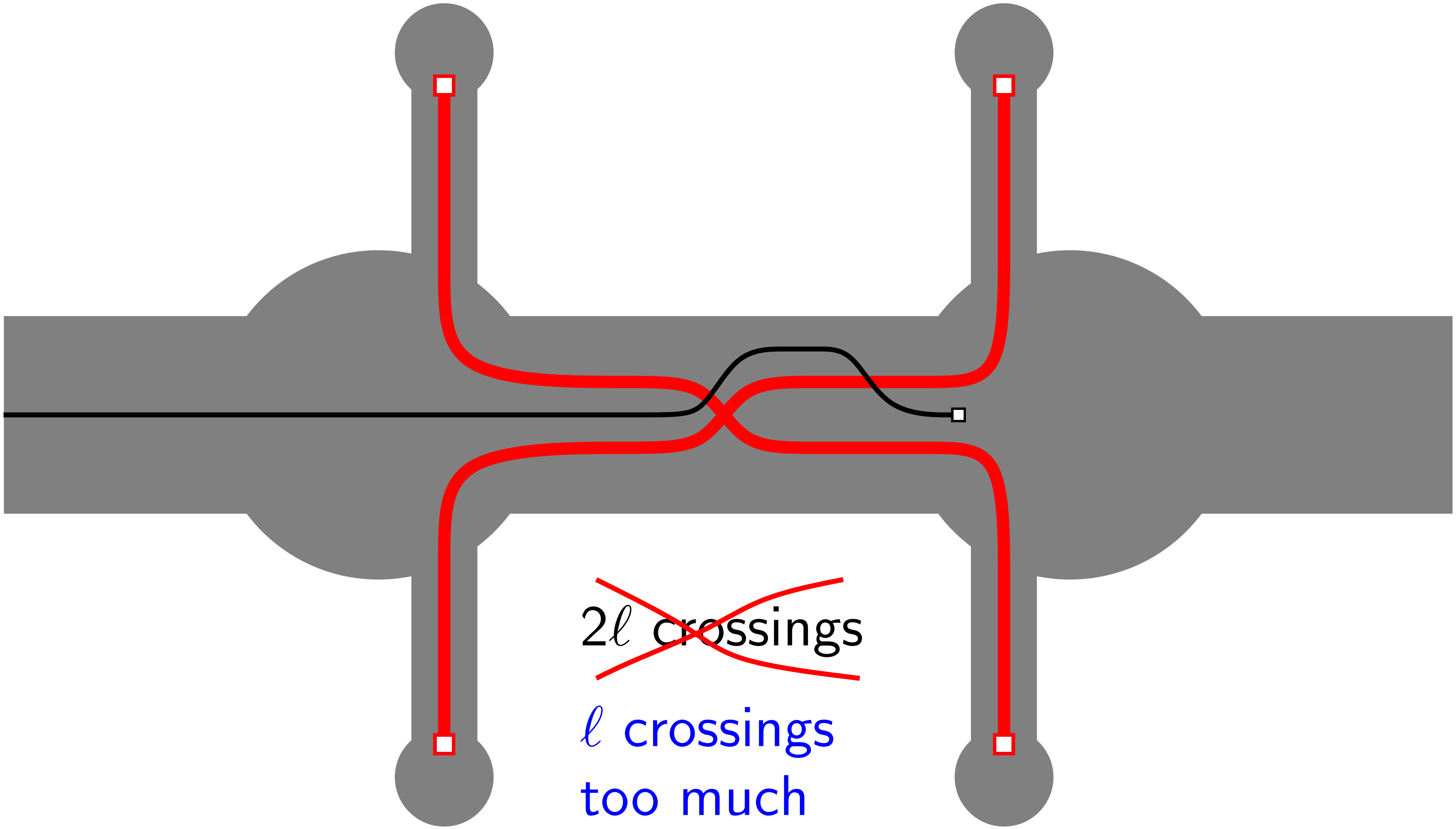
# MLCM – NP-hardness cont'd



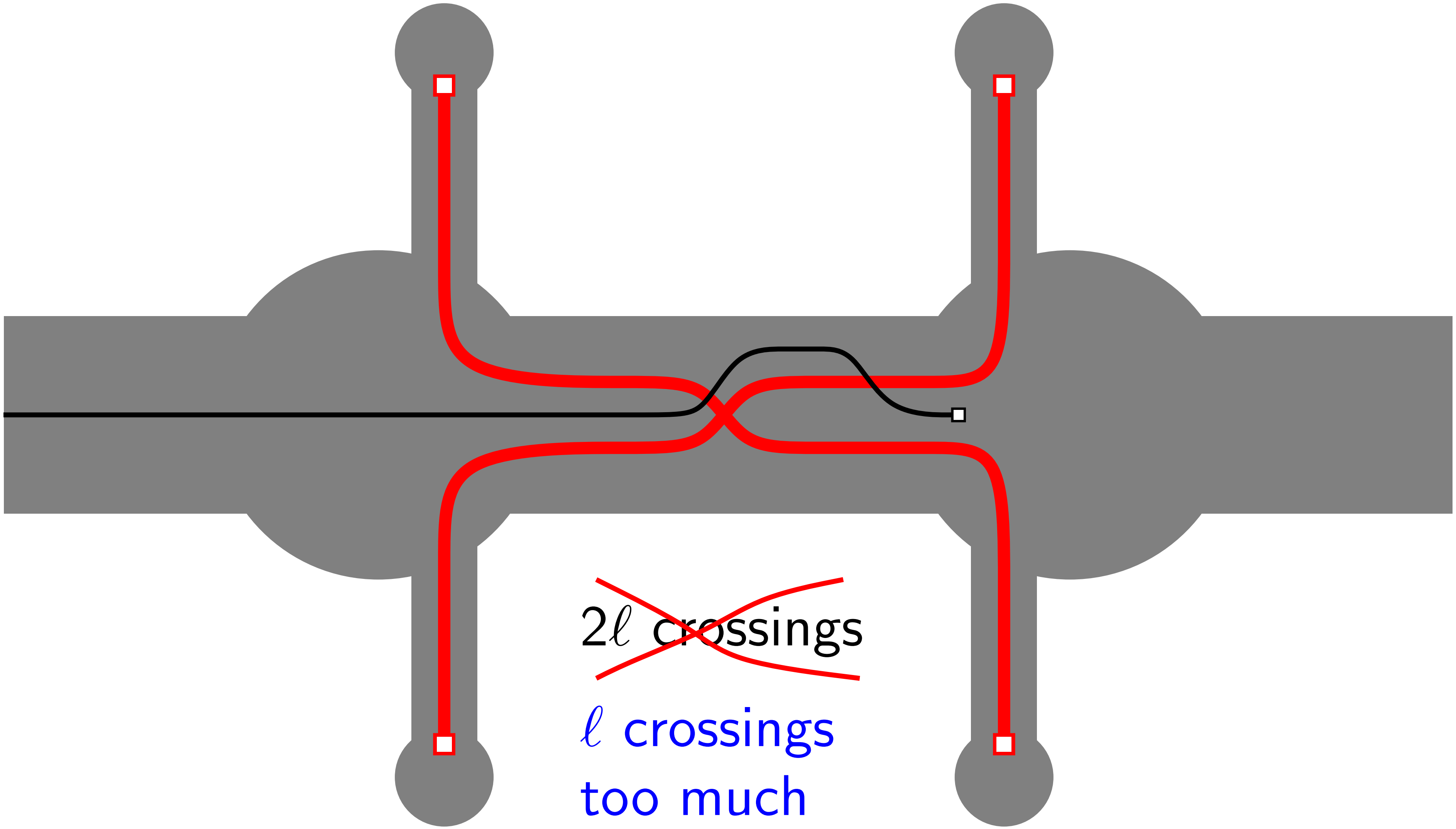
# MLCM – NP-hardness cont'd



# MLCM – NP-hardness cont'd

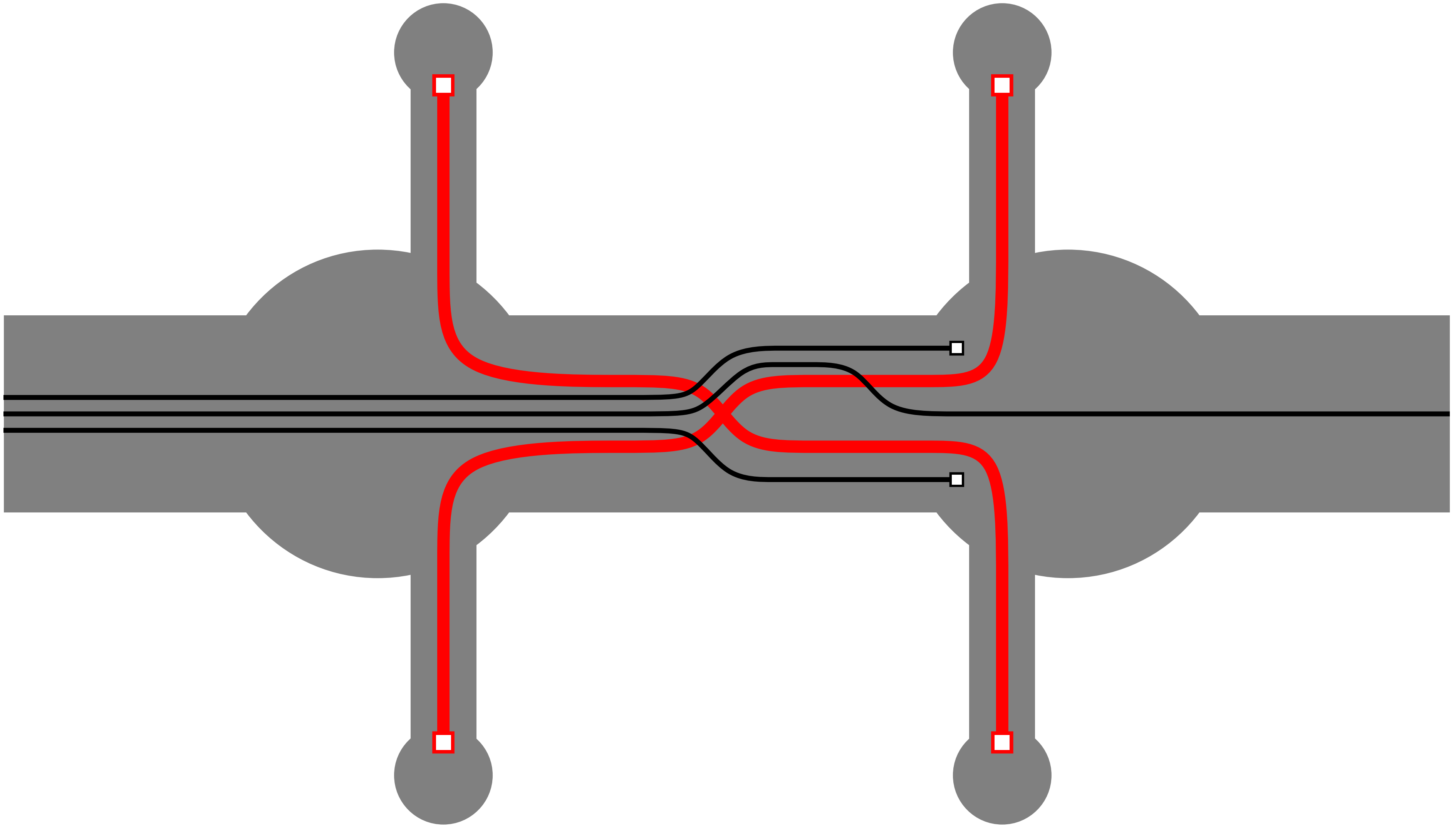


# MLCM – NP-hardness cont'd



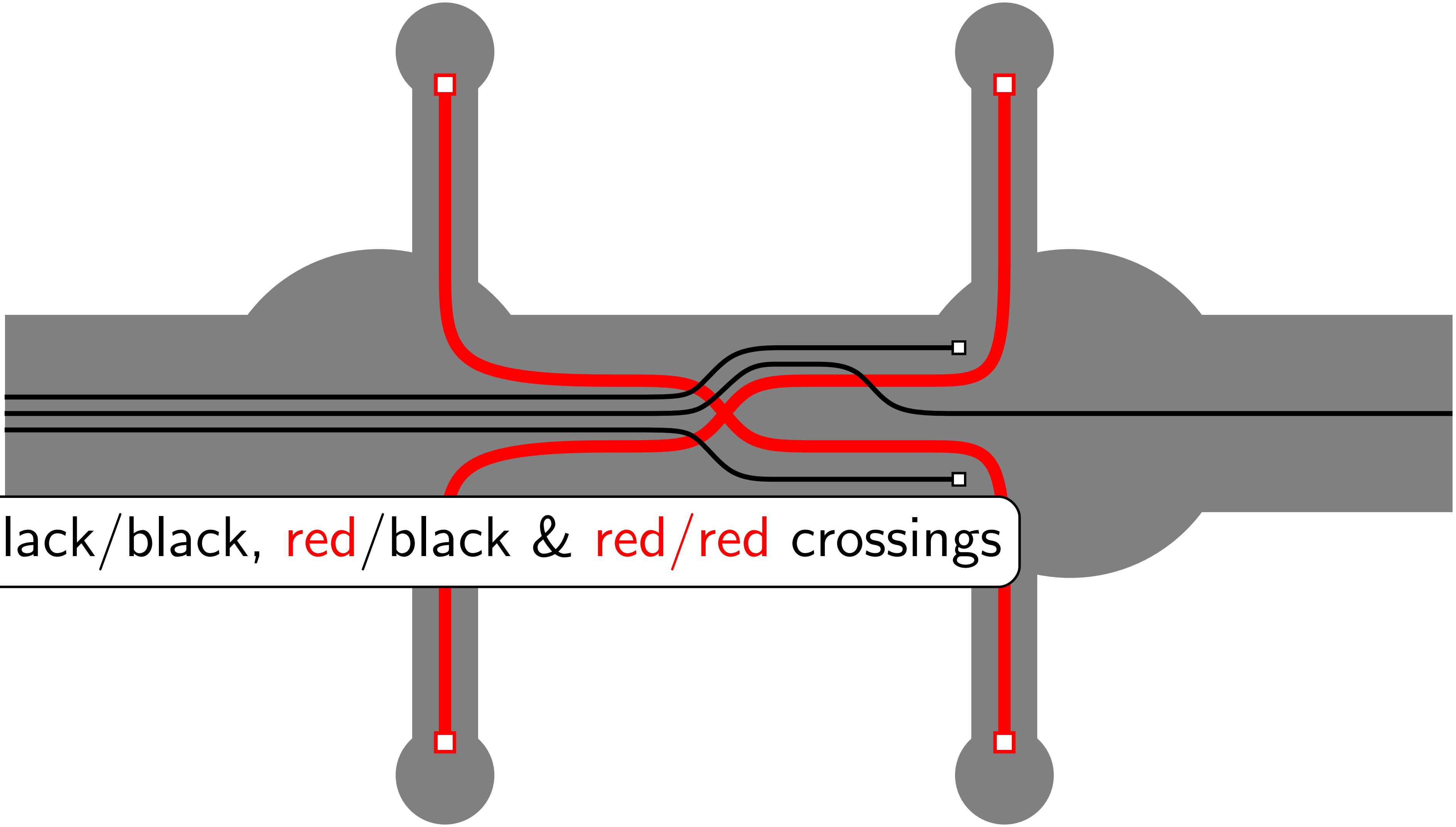

- good solutions are feasible for MLCM-P

# MLCM – NP-hardness cont'd



- good solutions are feasible for MLCM-P

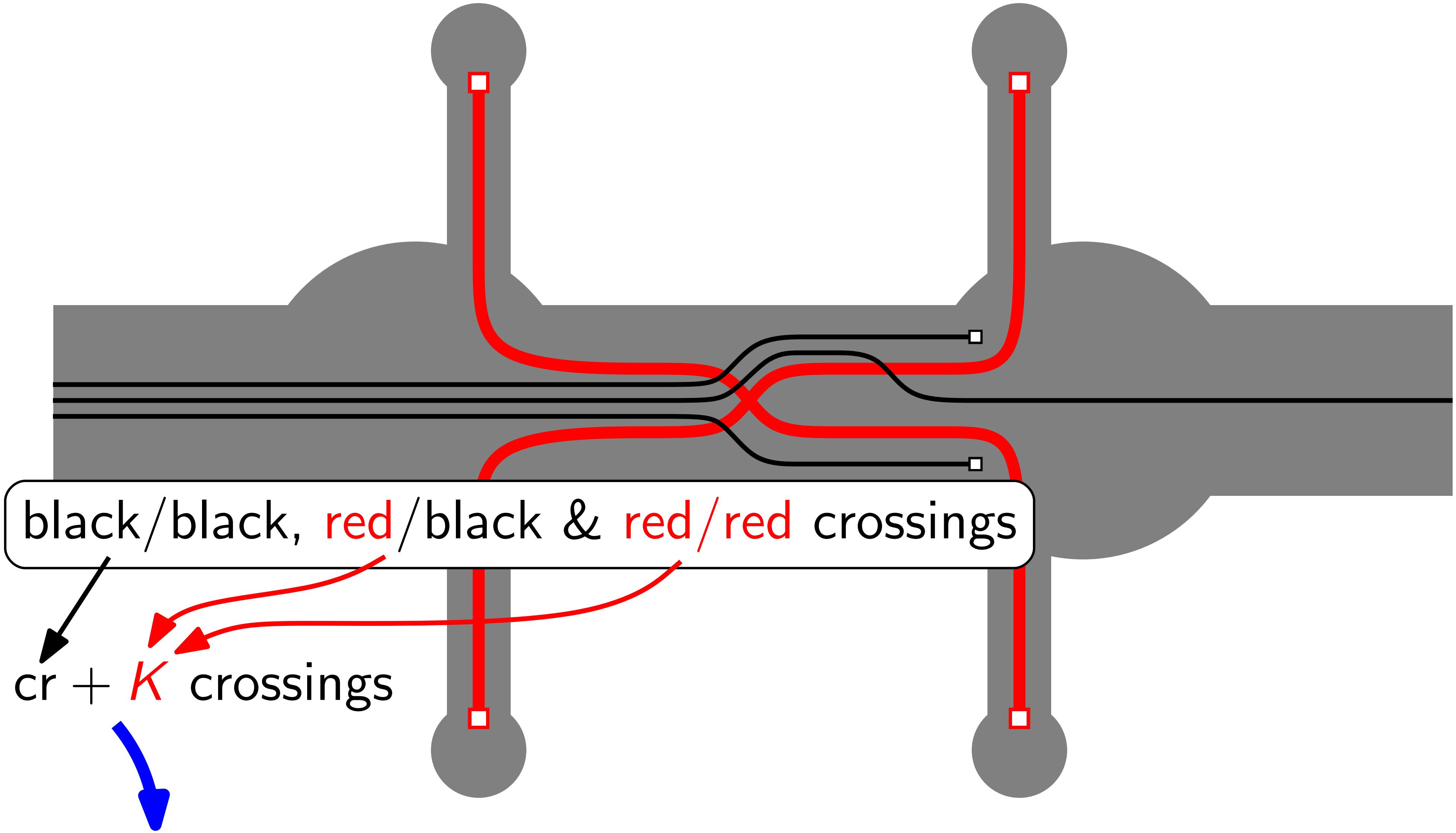
# MLCM – NP-hardness cont'd



black/black, red/black & red/red crossings

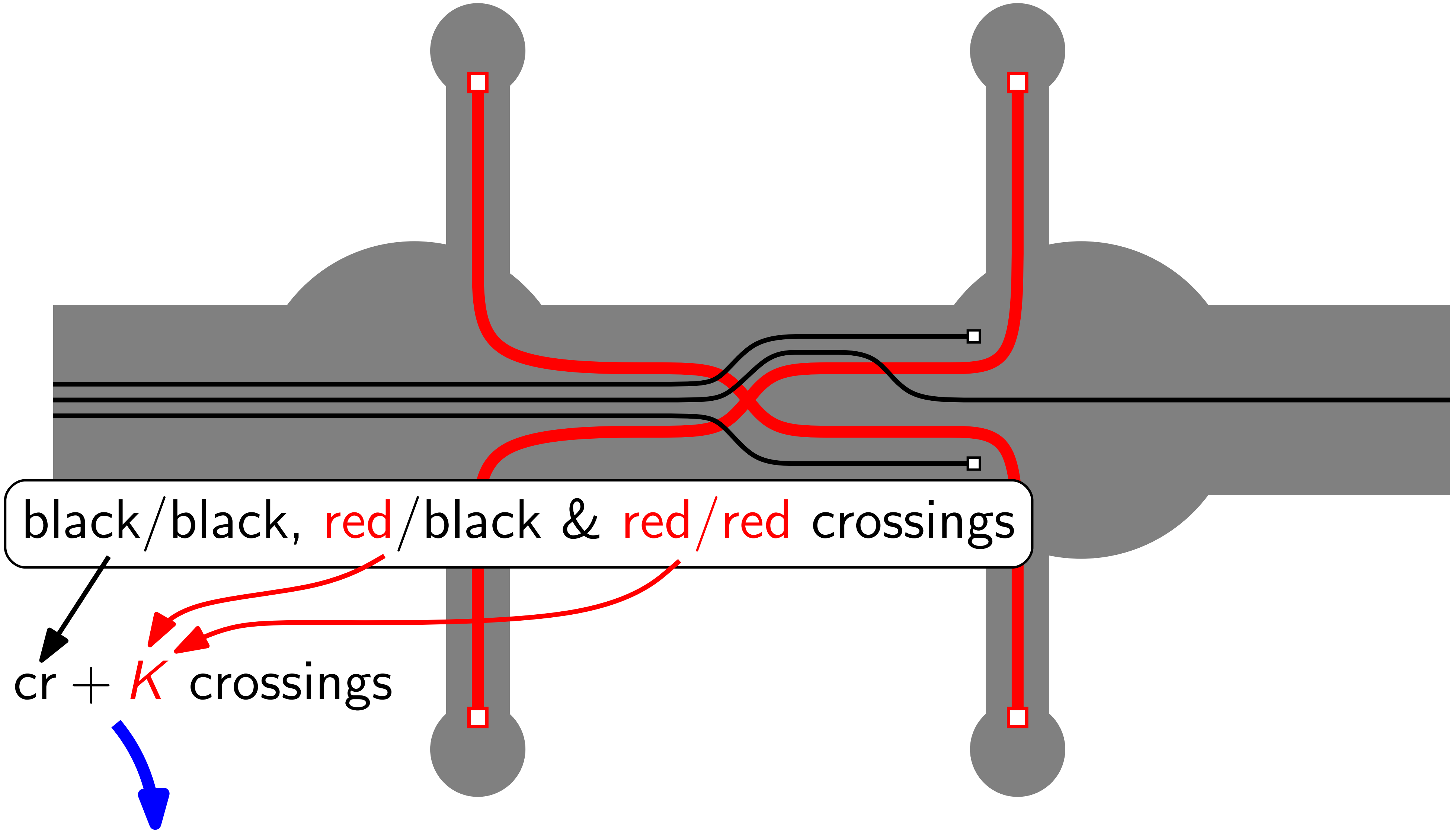
- good solutions are feasible for MLCM-P

# MLCM – NP-hardness cont'd



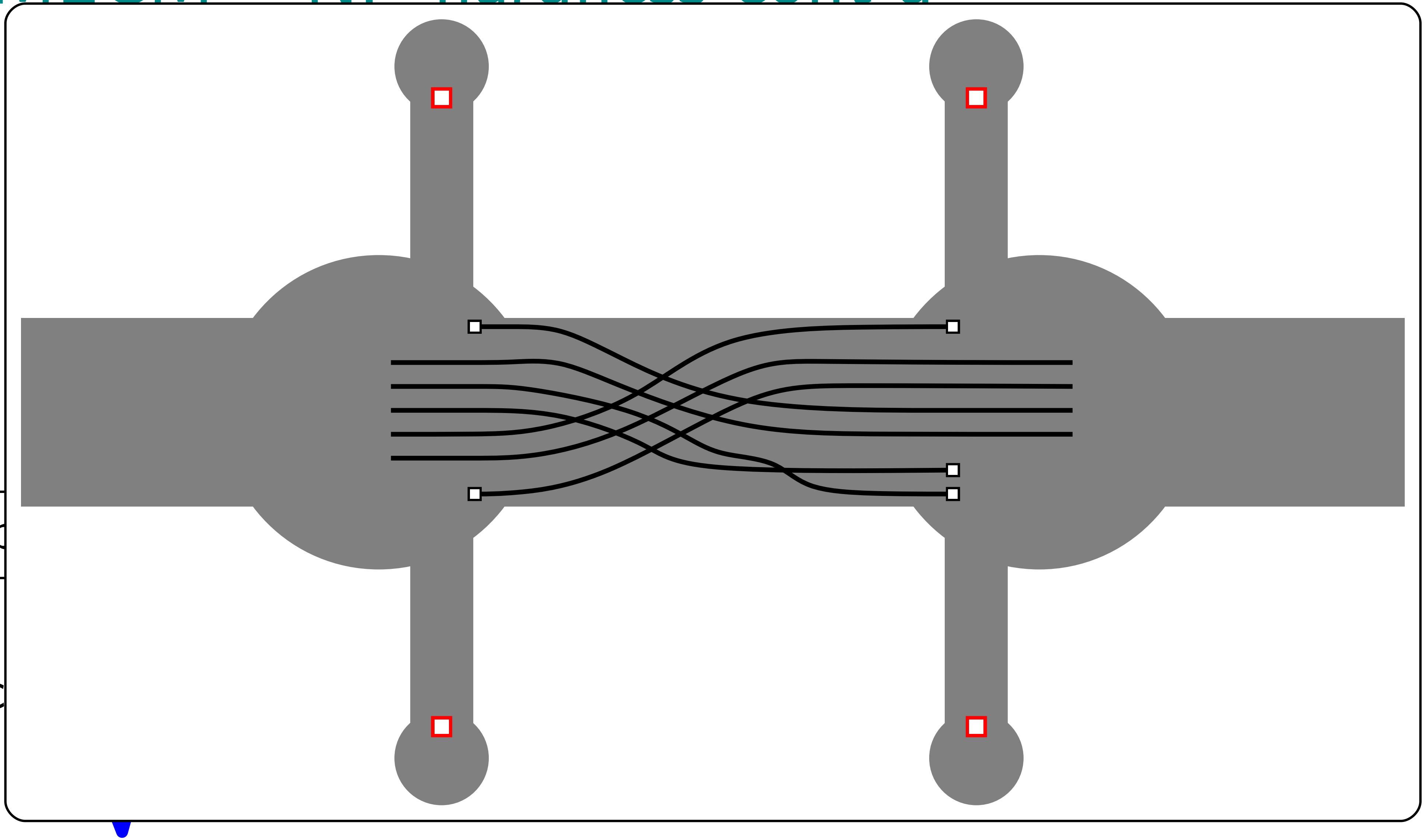

- good solutions are feasible for MLCM-P

# MLCM – NP-hardness cont'd



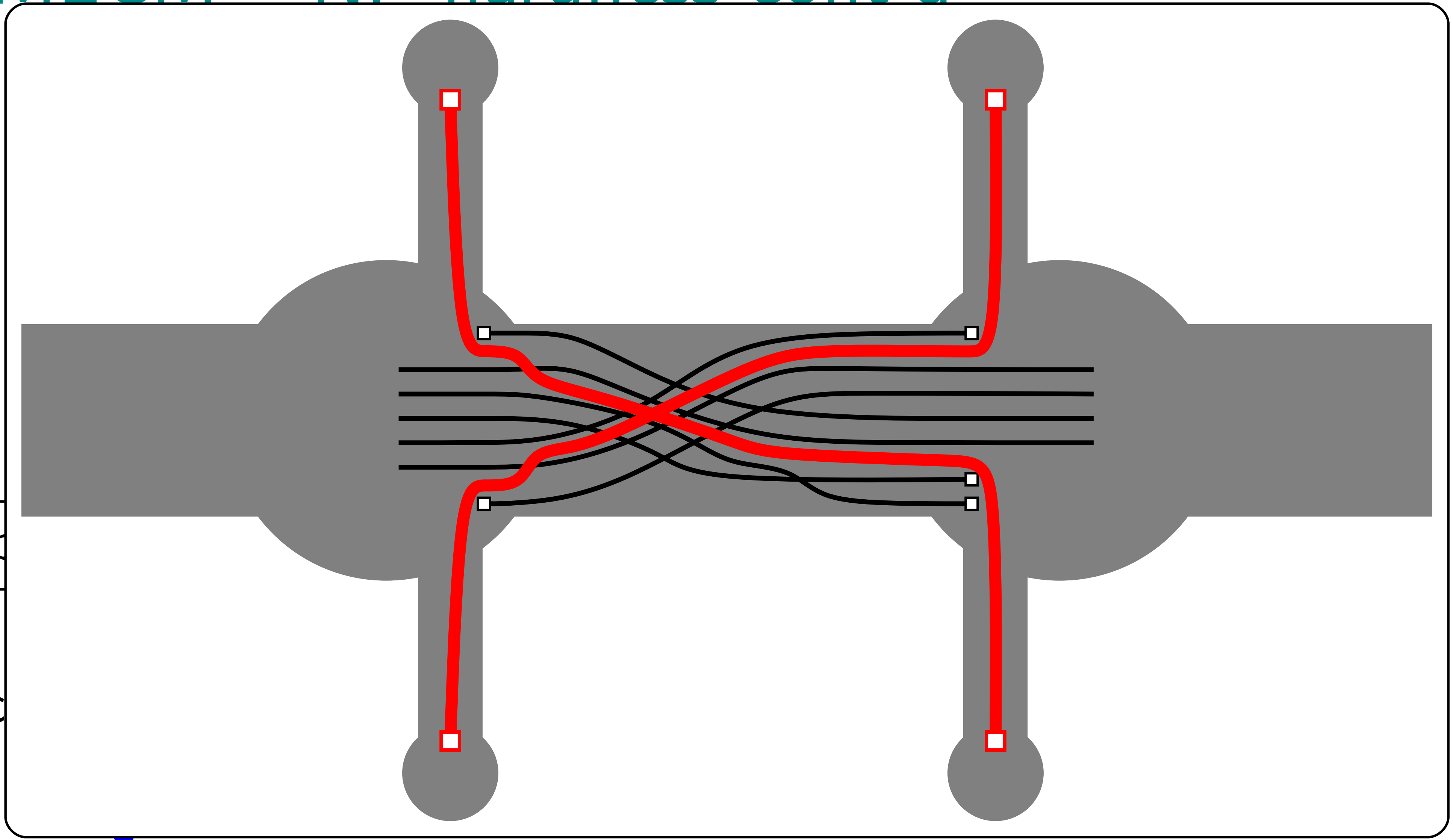

- good solutions are feasible for MLCM-P
- 1:1 correspondence of crossing-minimal solutions

# MLCM – NP-hardness cont'd



- good solutions are feasible for MLCM-P
- 1:1 correspondence of crossing-minimal solutions

# MLCM – NP-hardness cont'd

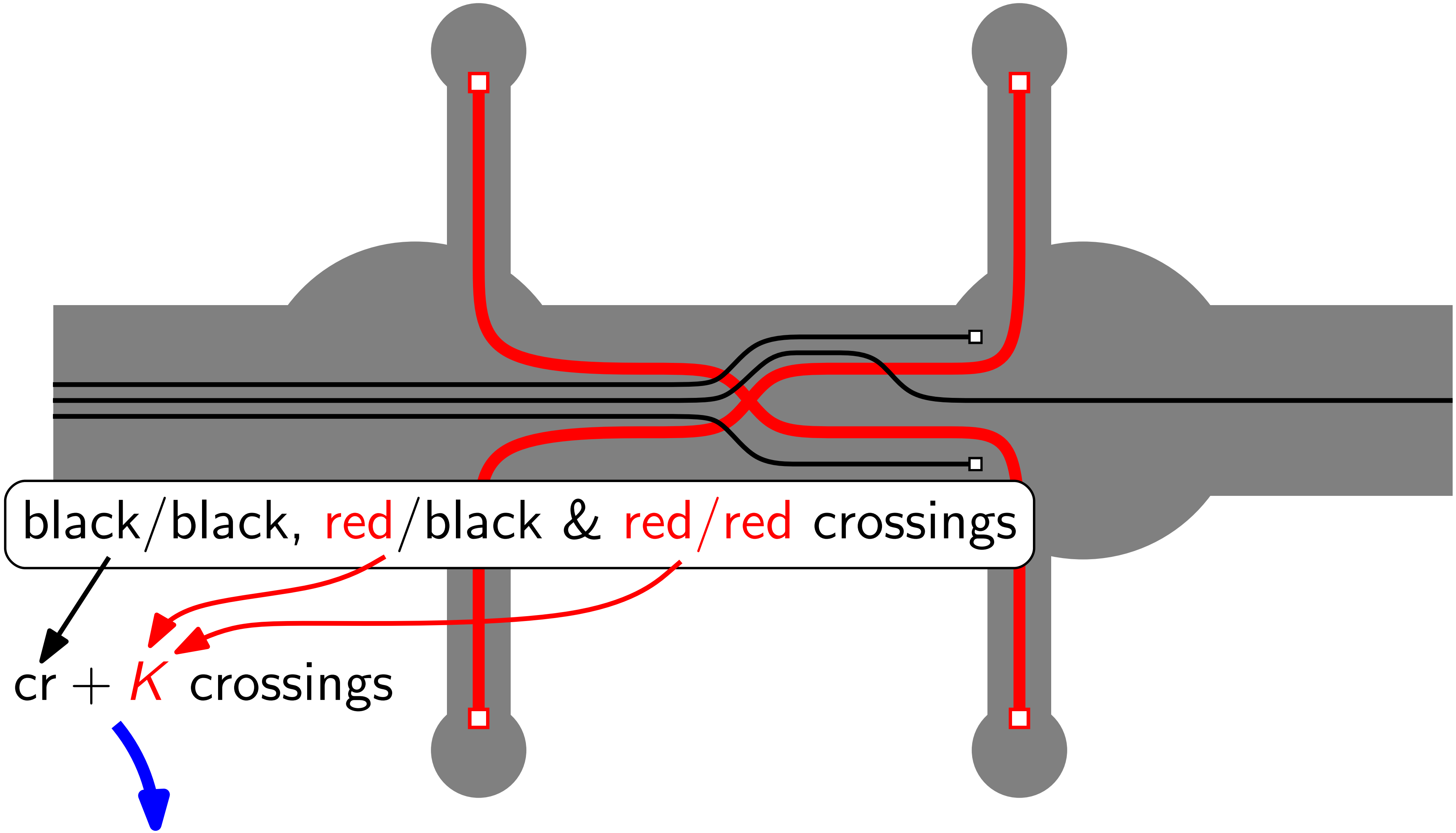


b

c

- good solutions are feasible for MLCM-P
- 1:1 correspondence of crossing-minimal solutions

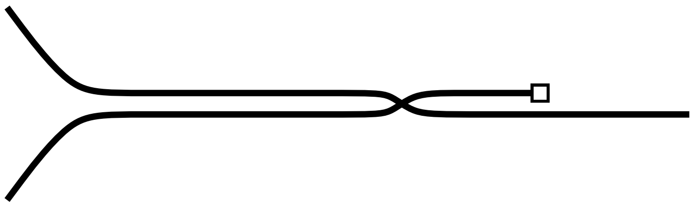
# MLCM – NP-hardness cont'd




- good solutions are feasible for MLCM-P
- 1:1 correspondence of crossing-minimal solutions

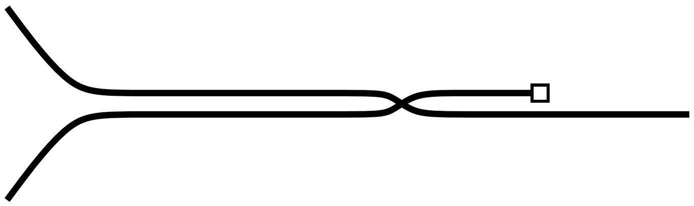
# MLCM – Crossing-free solutions

- avoidable crossings

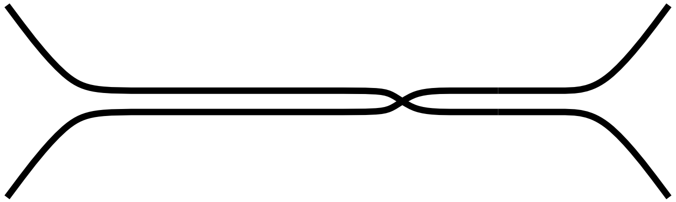


# MLCM – Crossing-free solutions

- avoidable crossings

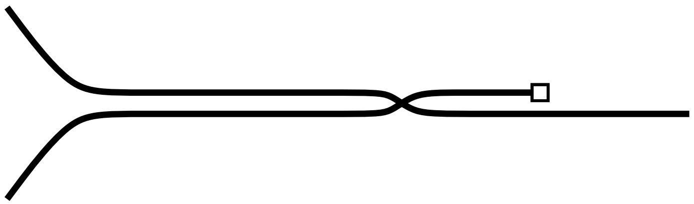


- unavoidable crossings

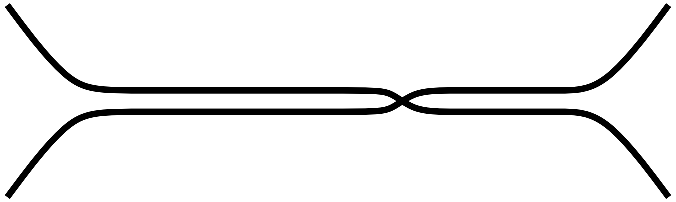


# MLCM – Crossing-free solutions

- avoidable crossings



- unavoidable crossings

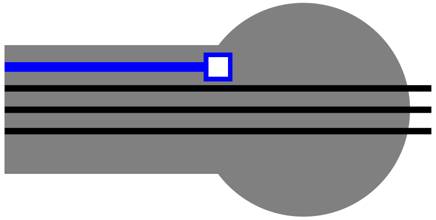


- We show:
  crossing-free solution exists $\Leftrightarrow$ no unavoidable crossing

- Main results:
  - **MLCM** is NP-hard.
  - $O(\sqrt{\log |L|})$-approximation for **MLCM-P**
- Additional results:
  - **MLCM** + **MLCM-P**: check for crossing-free solution
  - **MLCM-P**: Checking for solution with $\leq k$ crossings is fixed-parameter tractable.
  - a restricted version of MLCM-P can be solved efficiently

# MLCM-P – 2SAT model

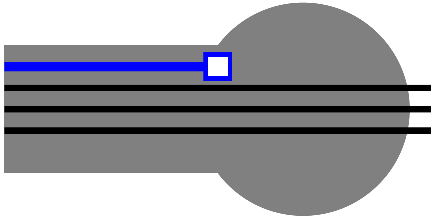
- variable $l_v$ models side of line end at node $v$

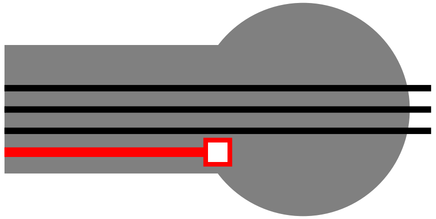


$l_v = \text{true}$

# MLCM-P – 2SAT model

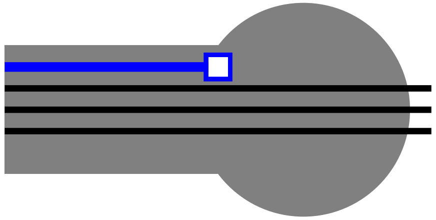
- variable $l_v$ models side of line end at node $v$



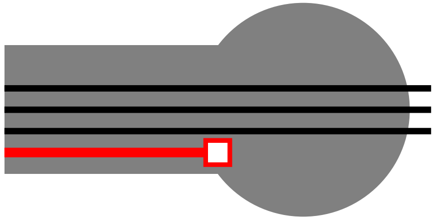



# MLCM-P – 2SAT model

- variable $l_v$ models side of line end at node $v$



$l_v = \text{true}$

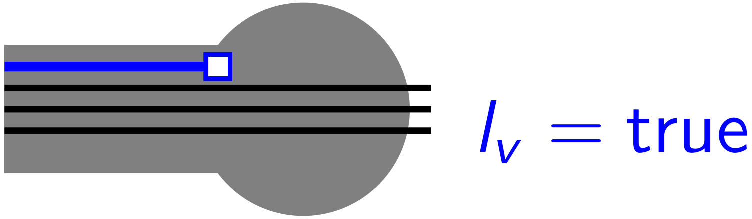


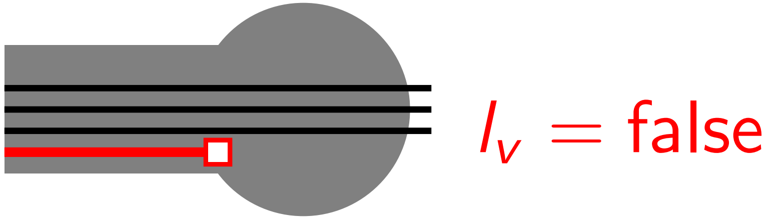
$l_v = \text{false}$

- analyze pairs of lines

# MLCM-P – 2SAT model

- variable $l_v$ models side of line end at node $v$



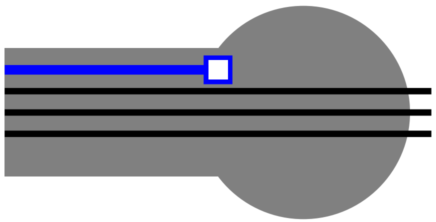




- analyze pairs of lines

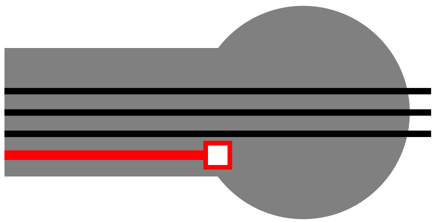

crossing can be avoided

# MLCM-P – 2SAT model

- variable $l_v$ models side of line end at node $v$

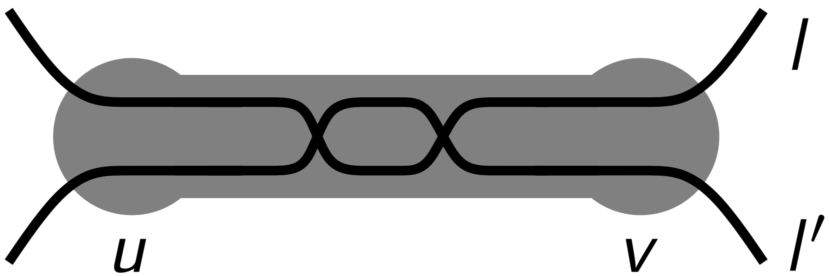


$l_v = \text{true}$



$l_v = \text{false}$

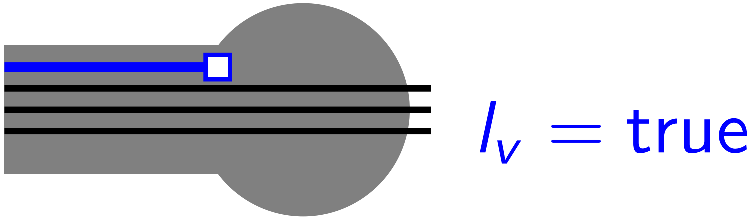
- analyze pairs of lines

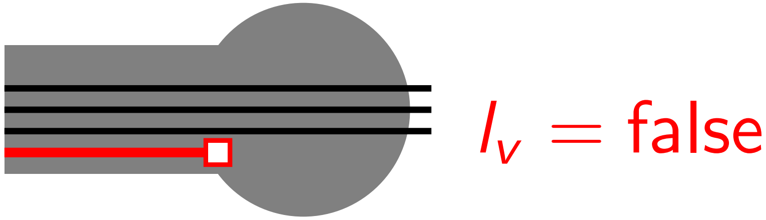



crossing can be avoided

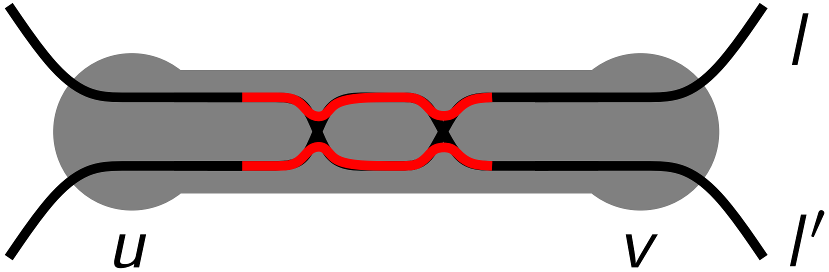
# MLCM-P – 2SAT model

- variable $l_v$ models side of line end at node $v$



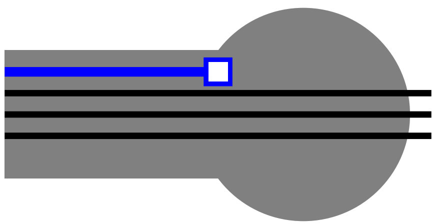

- analyze pairs of lines



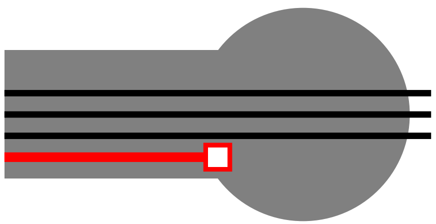

crossing can be avoided

# MLCM-P – 2SAT model

- variable $l_v$ models side of line end at node $v$

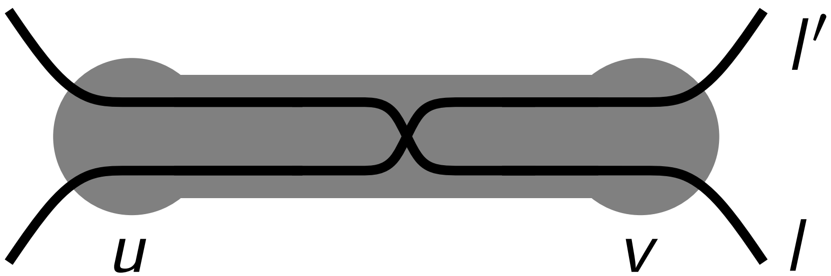


$l_v = \text{true}$



$l_v = \text{false}$

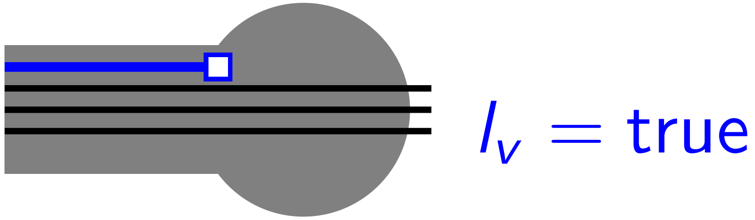
- analyze pairs of lines

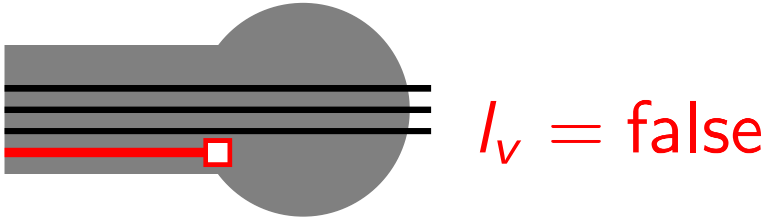



crossing cannot be avoided

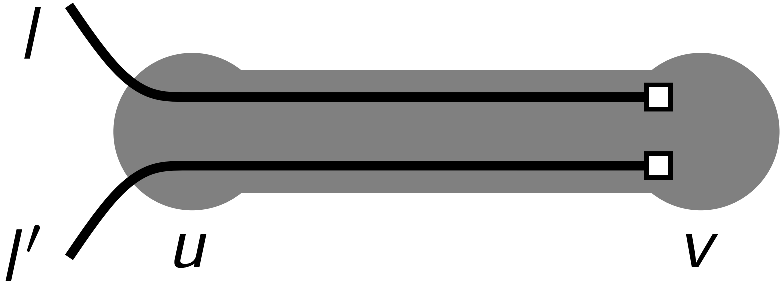
# MLCM-P – 2SAT model

- variable $l_v$ models side of line end at node $v$



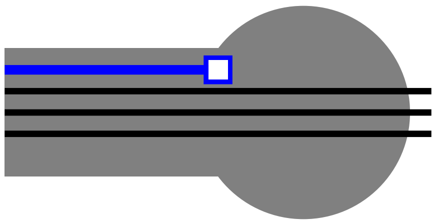




- analyze pairs of lines

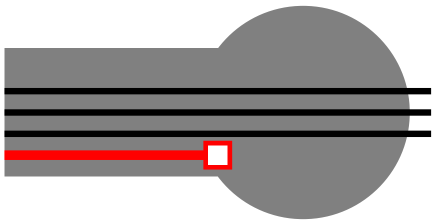


# MLCM-P – 2SAT model

- variable $l_v$ models side of line end at node $v$

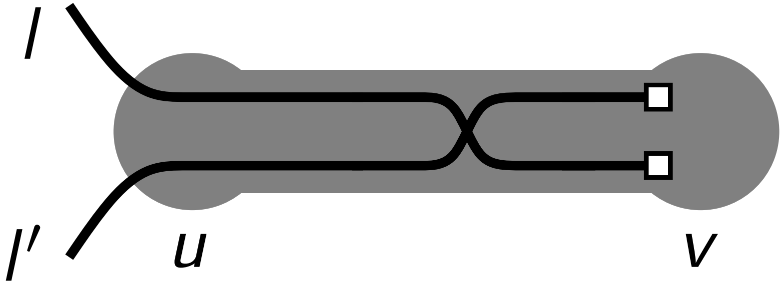


$l_v = \text{true}$



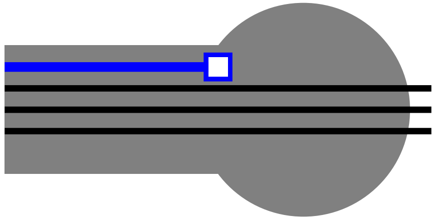
$l_v = \text{false}$

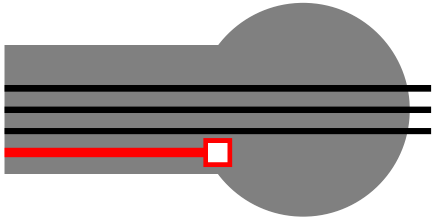
- analyze pairs of lines



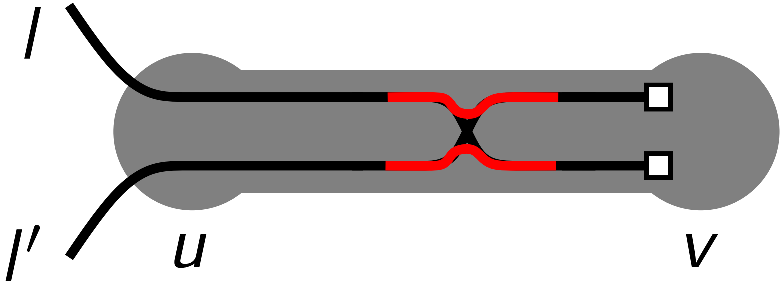
# MLCM-P – 2SAT model

- variable $l_v$ models side of line end at node $v$



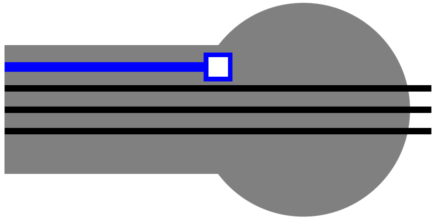

- analyze pairs of lines



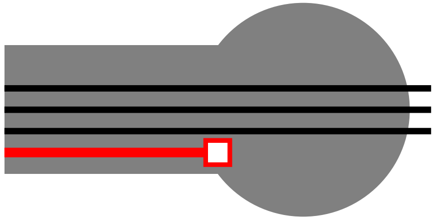

crossing can be avoided

# MLCM-P – 2SAT model

- variable $l_v$ models side of line end at node $v$

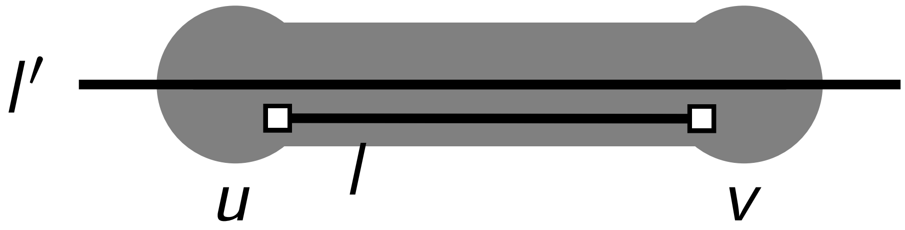


$l_v = \text{true}$



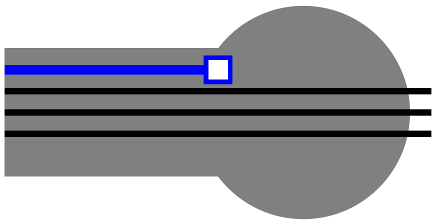
$l_v = \text{false}$

- analyze pairs of lines

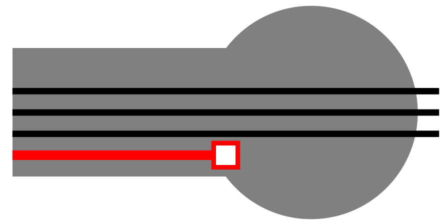


# MLCM-P – 2SAT model

- variable $l_v$ models side of line end at node $v$

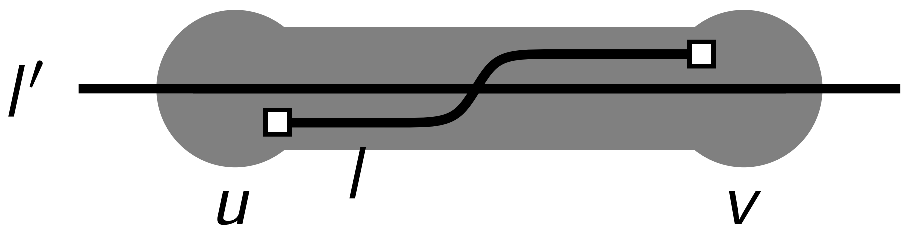


$l_v = \text{true}$



$l_v = \text{false}$

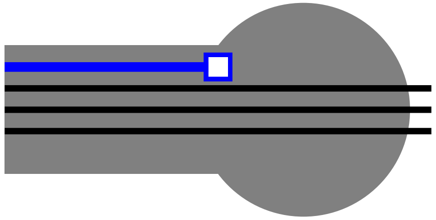
- analyze pairs of lines

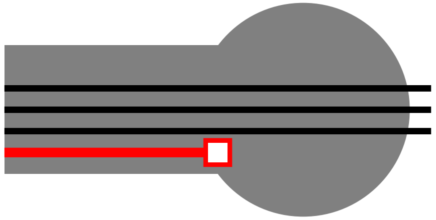



no crossing $\Leftrightarrow l_u \equiv l_v$

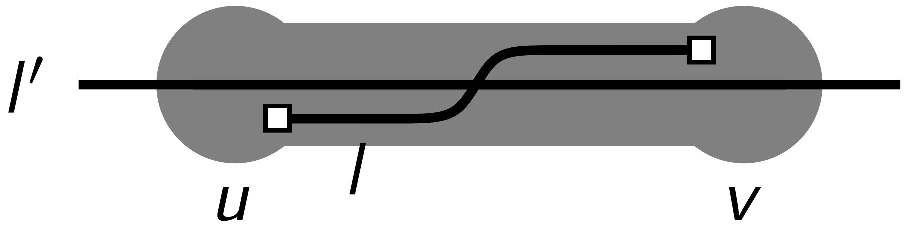
# MLCM-P – 2SAT model

- variable $l_v$ models side of line end at node $v$




- analyze pairs of lines

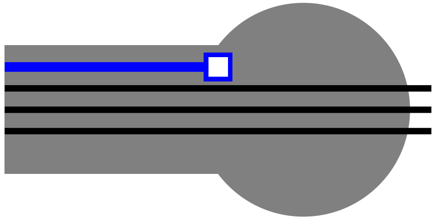



no crossing $\Leftrightarrow l_u \equiv l_v$

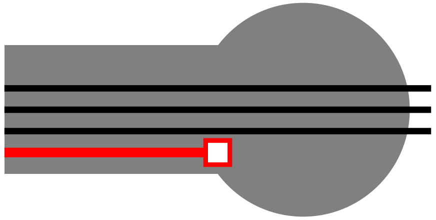
$\leftrightarrow (l_u \vee \neg l_v) \wedge (\neg l_u \vee l_v)$

# MLCM-P – 2SAT model

- variable $l_v$ models side of line end at node $v$

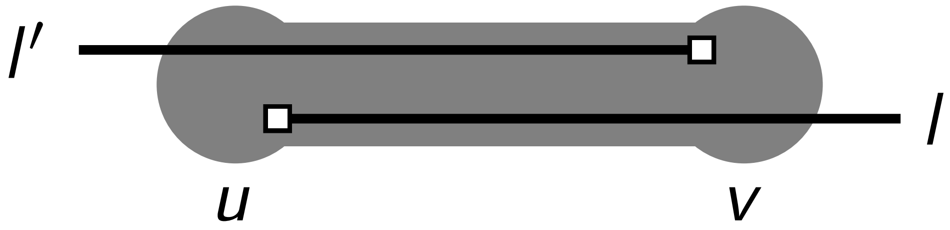


$l_v = \text{true}$



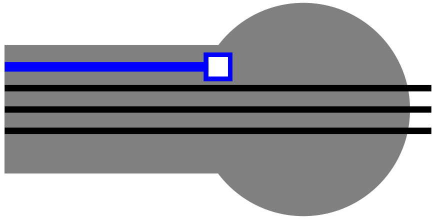
$l_v = \text{false}$

- analyze pairs of lines

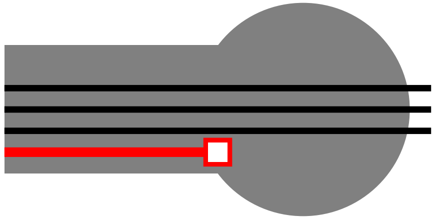


# MLCM-P – 2SAT model

- variable $l_v$ models side of line end at node $v$

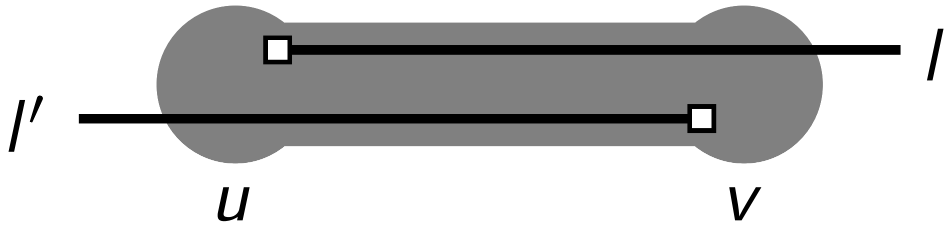


$l_v = \text{true}$



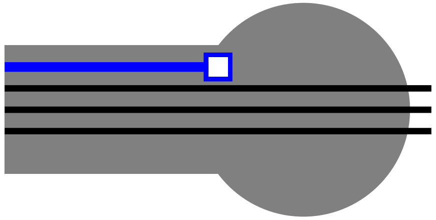
$l_v = \text{false}$

- analyze pairs of lines

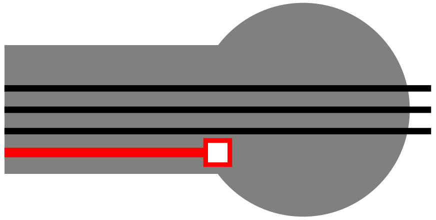


# MLCM-P – 2SAT model

- variable $l_v$ models side of line end at node $v$

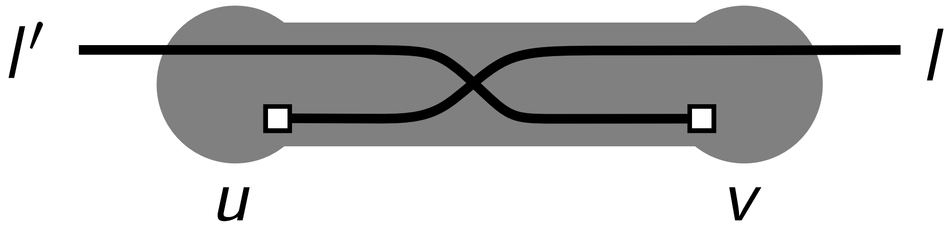


$l_v = \text{true}$



$l_v = \text{false}$

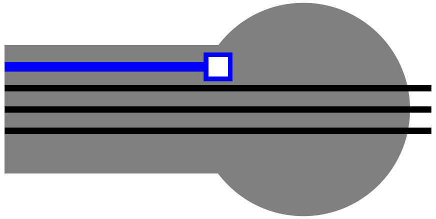
- analyze pairs of lines



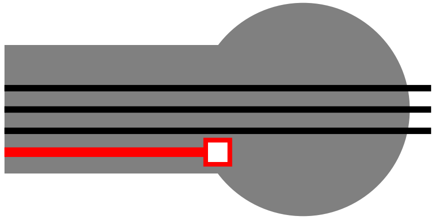

no crossing $\Leftrightarrow l_u \not\equiv l'_v$

# MLCM-P – 2SAT model

- variable $l_v$ models side of line end at node $v$

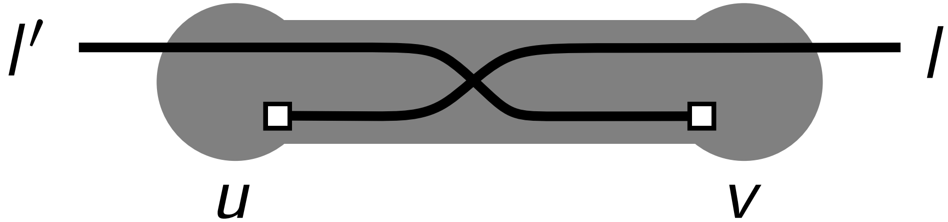


$l_v = \text{true}$



$l_v = \text{false}$

- analyze pairs of lines

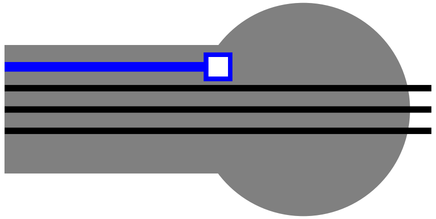



no crossing $\Leftrightarrow l_u \not\equiv l'_v$

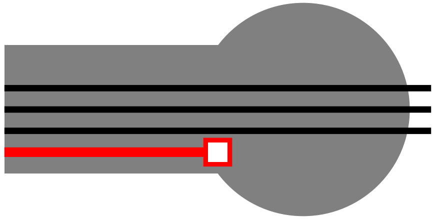
$\leftrightarrow (l_u \vee l'_v) \wedge (\neg l_u \vee \neg l'_v)$

# MLCM-P – 2SAT model

- variable $l_v$ models side of line end at node $v$



$l_v = \text{true}$

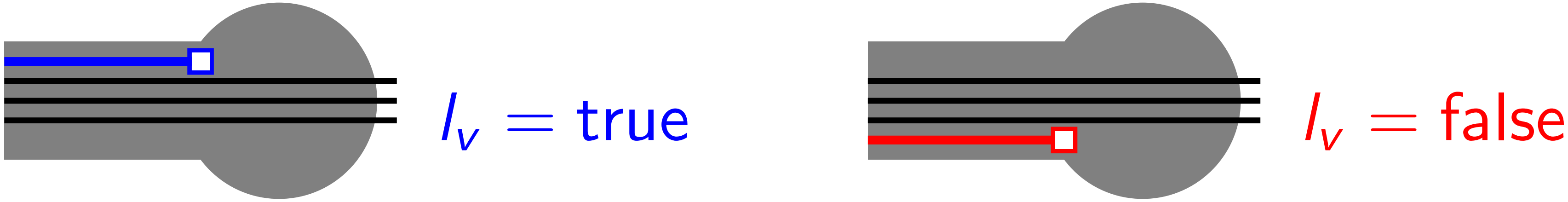


$l_v = \text{false}$

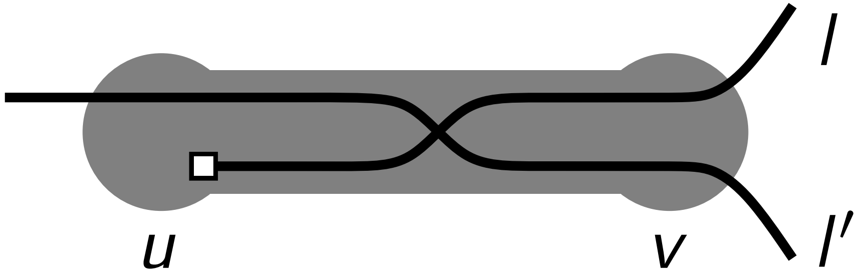
- analyze pairs of lines

# MLCM-P – 2SAT model

- variable $l_v$ models side of line end at node $v$



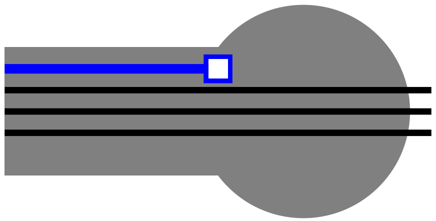

- analyze pairs of lines



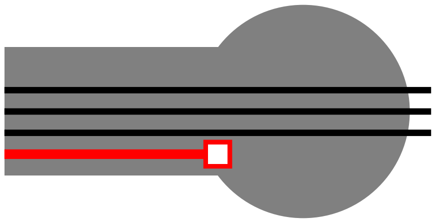

no crossing $\Leftrightarrow l_u \equiv \text{true}$

# MLCM-P – 2SAT model

- variable $l_v$ models side of line end at node $v$

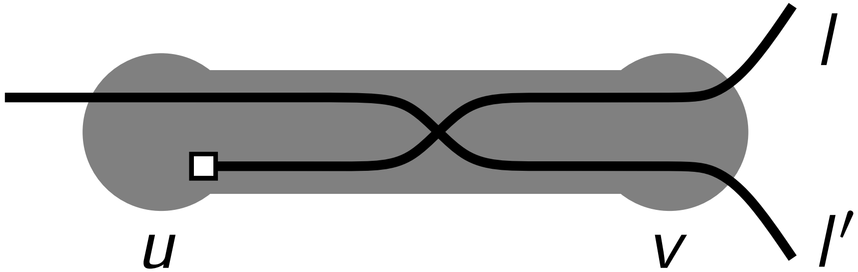


$l_v = \text{true}$



$l_v = \text{false}$

- analyze pairs of lines

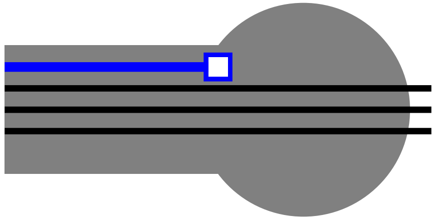



no crossing $\Leftrightarrow l_u \equiv \text{true}$

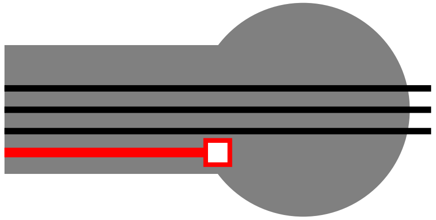
$\leftrightarrow (l_u)$

# MLCM-P – 2SAT model

- variable $l_v$ models side of line end at node $v$

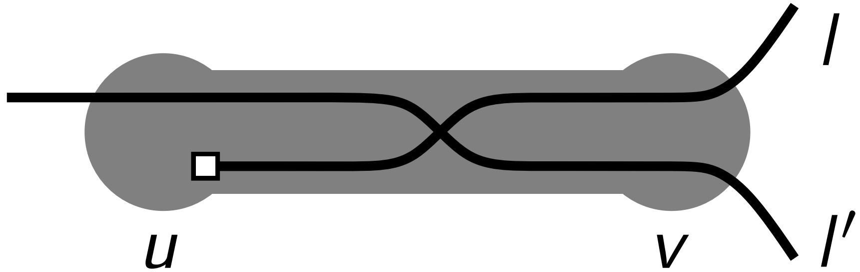


$l_v = \text{true}$



$l_v = \text{false}$

- analyze pairs of lines




no crossing $\Leftrightarrow l_u \equiv \text{true}$

$\leftrightarrow (l_u)$

- checking for crossing-free solution in $O(|L|^2)$ time

note: replace $l_u$ by $\neg l_u$ if necessary for consistent meaning of "top"

# MLCM-P – Approximation

- check for unavoidable crossings
- build 2SAT model

# MLCM-P – Approximation

- check for unavoidable crossings
- build 2SAT model
- $k$ avoidable crossings $\leftrightarrow$ $k$ unsatisfied crossing formulas

# MLCM-P – Approximation

- check for unavoidable crossings
- build 2SAT model
- $k$ avoidable crossings $\leftrightarrow$ $k$ unsatisfied crossing formulas

$$(\mathbf{l_u} \vee \neg l_v) \wedge (\neg \mathbf{l_u} \vee l_v)$$

at least one clause satisfied

# MLCM-P – Approximation

- check for unavoidable crossings
- build 2SAT model
- $k$ avoidable crossings $\leftrightarrow$ $k$ unsatisfied crossing formulas

$$(\mathbf{l_u} \vee \neg l_v) \wedge (\neg \mathbf{l_u} \vee l_v)$$

at least one clause satisfied

- $k$ avoidable crossings $\leftrightarrow$ $k$ unsatisfied clauses

# MLCM-P – Approximation

- check for unavoidable crossings
- build 2SAT model
- $k$ avoidable crossings $\leftrightarrow$ $k$ unsatisfied crossing formulas

$$(\mathbf{l_u} \vee \neg l_v) \wedge (\neg \mathbf{l_u} \vee l_v)$$

at least one clause satisfied

- $k$ avoidable crossings $\leftrightarrow$ $k$ unsatisfied clauses
- $O(\sqrt{\log n})$-approximation for **Min 2CNF Deletion** $\Rightarrow$ $O(\sqrt{\log |L|})$-approximation for **MLCM-P**

# MLCM-P – Approximation

- check for unavoidable crossings
- build 2SAT model
- $k$ avoidable crossings $\leftrightarrow$ $k$ unsatisfied crossing formulas

$$(\mathbf{l_u} \vee \neg l_v) \wedge (\neg \mathbf{l_u} \vee l_v)$$

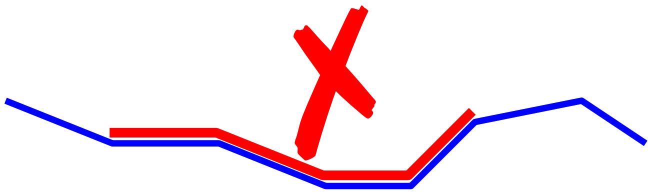
at least one clause satisfied

- $k$ avoidable crossings $\leftrightarrow$ $k$ unsatisfied clauses
- $O(\sqrt{\log n})$-approximation for **Min 2CNF Deletion** $\Rightarrow$ $O(\sqrt{\log |L|})$-approximation for **MLCM-P**
- similarly: MLCM-P is fixed-parameter tractable.

- Main results:
  - **MLCM** is NP-hard.
  - $O(\sqrt{\log |L|})$-approximation for **MLCM-P**
- Additional results:
  - **MLCM** + **MLCM-P**: check for crossing-free solution
  - **MLCM-P**: Checking for solution with $\leq k$ crossings is fixed-parameter tractable.
  - a restricted version of MLCM-P can be solved efficiently

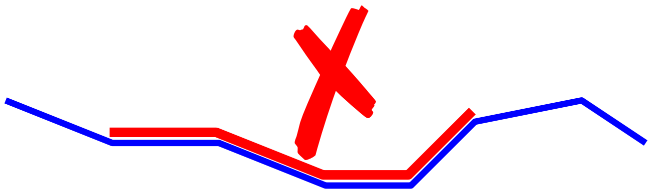
# Proper MLCM-P

- no line is subpath of another line

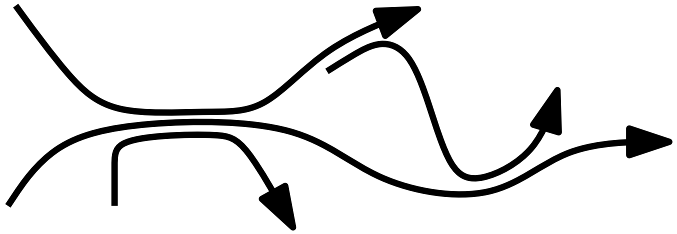


# Proper MLCM-P

- no line is subpath of another line

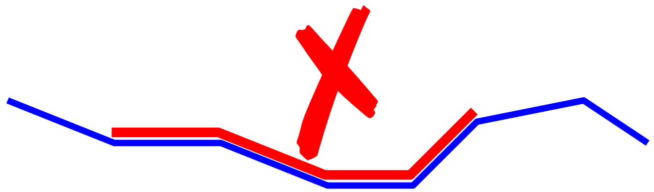


- consistent line directions

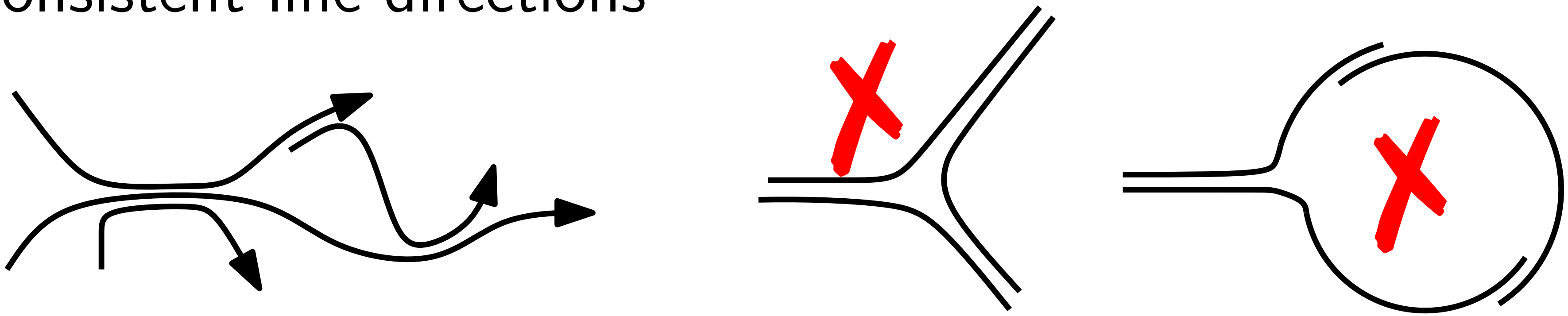


# Proper MLCM-P

- no line is subpath of another line

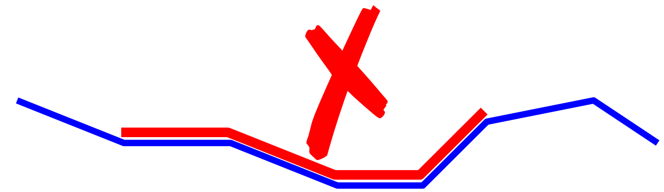


- consistent line directions

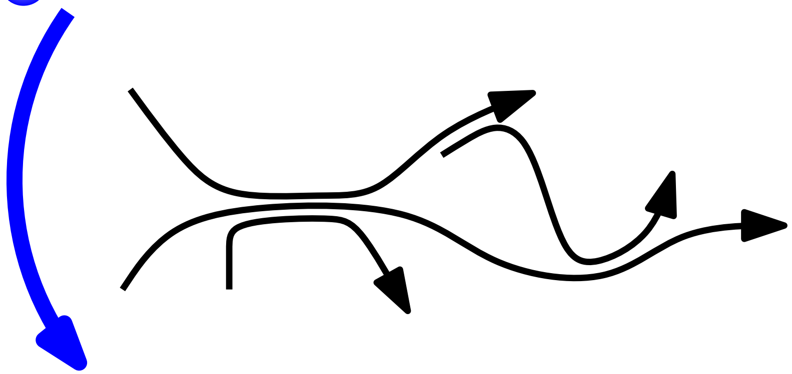


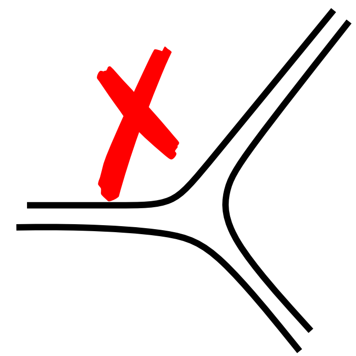
# Proper MLCM-P

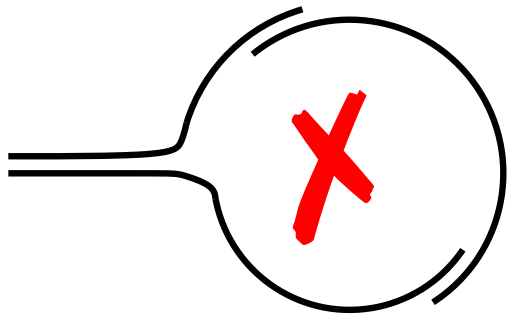
- no line is subpath of another line



- consistent line directions



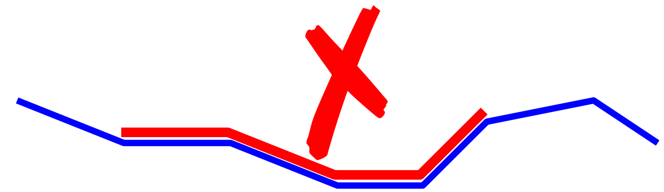




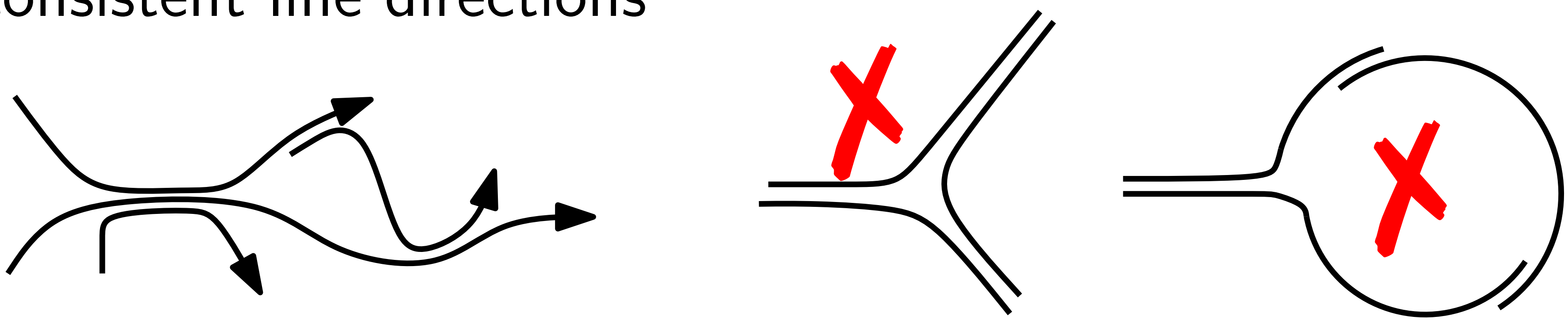
- global “top” / “bottom”

# Proper MLCM-P

- no line is subpath of another line



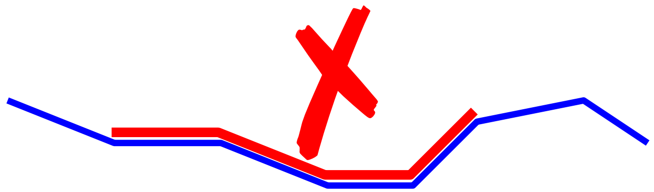
- consistent line directions



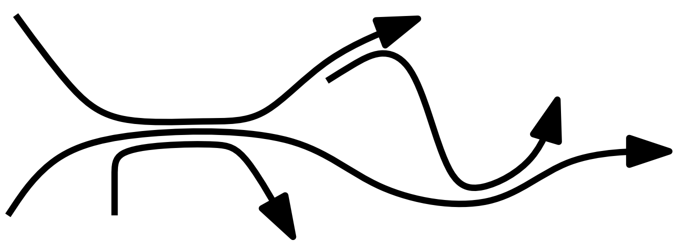
- global "top" / "bottom"
- no equality crossing formulas $l_u \equiv l_v$

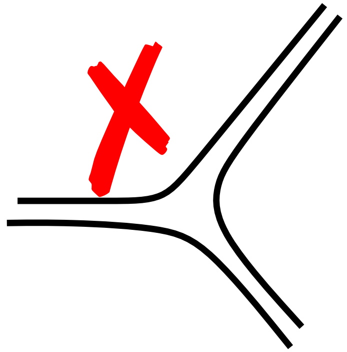
# Proper MLCM-P

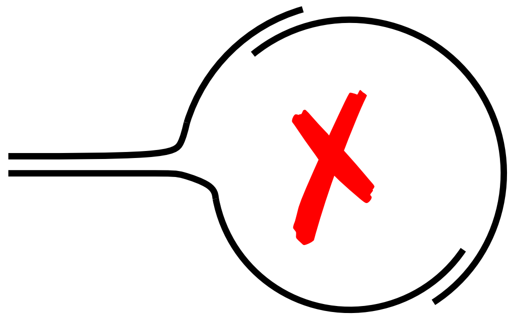
- no line is subpath of another line



- consistent line directions



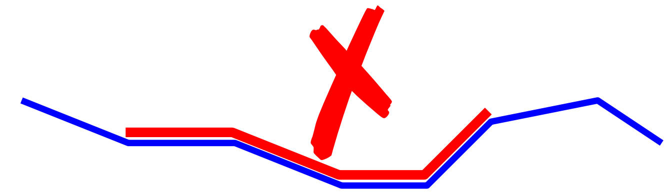




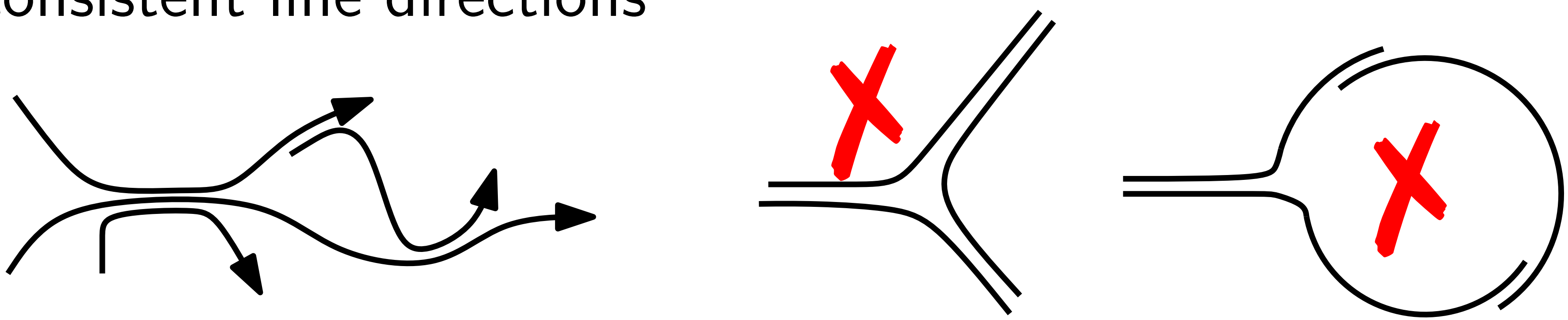
- global "top" / "bottom"
- no equality crossing formulas $l_u \equiv l_v$
- can solve Min 2CNF Deletion efficiently by MIN UNCUT model

# Proper MLCM-P

- no line is subpath of another line



- consistent line directions



- global "top" / "bottom"
- no equality crossing formulas $l_u \equiv l_v$
- can solve Min 2CNF Deletion efficiently by MIN UNCUT model
- $\Rightarrow$ polynomial-time algorithm for Proper MLCM-P

# Conclusion & Open Questions

- **MLCM** is NP-hard.
- $O(\sqrt{\log |L|})$-approximation for **MLCM-P**
- Proper MLCM-P with consistent directions can be solved efficiently.

# Conclusion & Open Questions

- **MLCM** is NP-hard.
- $O(\sqrt{\log |L|})$-approximation for **MLCM-P**
- Proper MLCM-P with consistent directions can be solved efficiently.

Open questions:

- approximation for MLCM
- constant-factor approximation for MLCM-P
- FPT for $\#$lines per edge

# Conclusion & Open Questions

- **MLCM** is NP-hard.
- $O(\sqrt{\log |L|})$-approximation for **MLCM-P**
- Proper MLCM-P with consistent directions can be solved efficiently.

Open questions:

- approximation for MLCM
- constant-factor approximation for MLCM-P
- FPT for #lines per edge

Thank you!